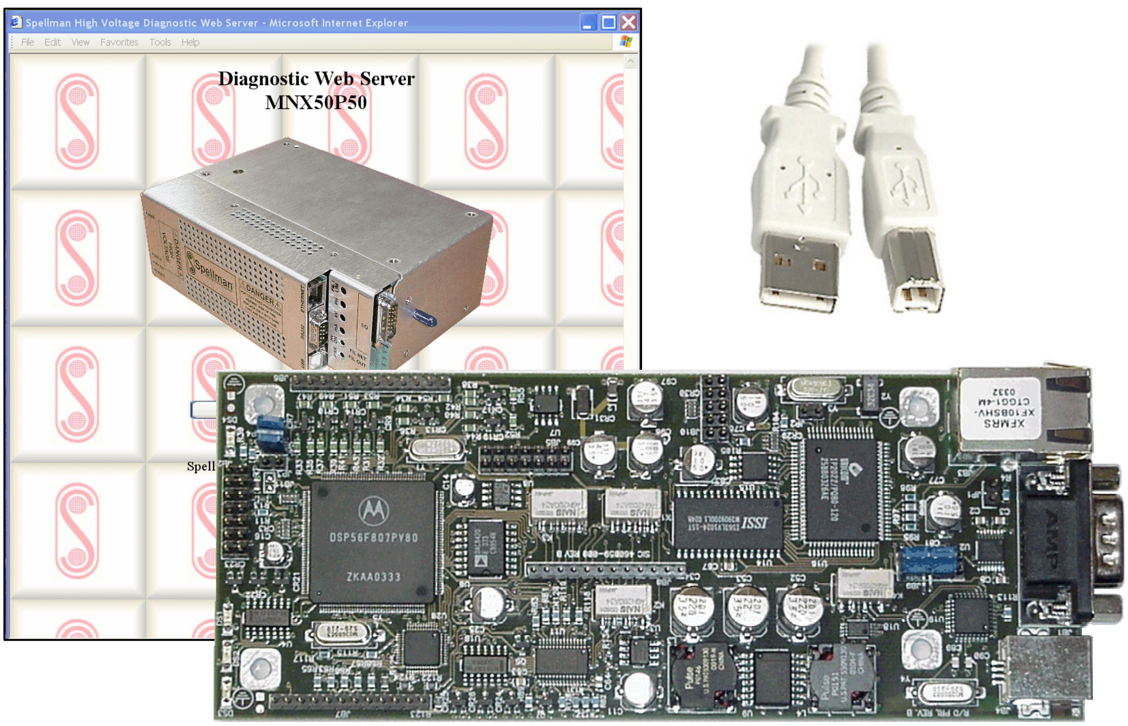
Standard Interface Control Option
SIC Digital Interface:

Ethernet
Serial – RS-232
Universal Serial Bus - USB



USA EUROPE JAPAN MEXICO

475 Wireless Boulevard • Hauppauge, New York 11788, USA • www.spellmanhv.com • T:+1 631.630.3000 • F:+1 631.435.1620

Changes

REVISION	DATE	DESCRIPTION
A	1/13/04	Draft version
B	2/11/04	Added enhanced 'Getting Started' sections
C	3/16/04	Added Sections 8.3 and 8.4

Table Of Contents

1.0 SCOPE

This document applies to the communications interfaces on the Standard Interface Control Board (SIC) Option, assembly 460059.

2.0 FUNCTIONAL DESCRIPTION

The SIC option provides 3 different types of communications interfaces:

- RS-232 on JB1
- Ethernet (10-Base-T) on JB3
- Universal Serial Bus on JB4.

Data acquisition and control capabilities are provided by:

- 14 channels of 12-bit analog-to-digital converters
- 2 additional analog channels that monitor the house-keeping power supply and ambient temperature
- 5 digital output bits
- 8 digital inputs bits
- 3 relays/interlocks

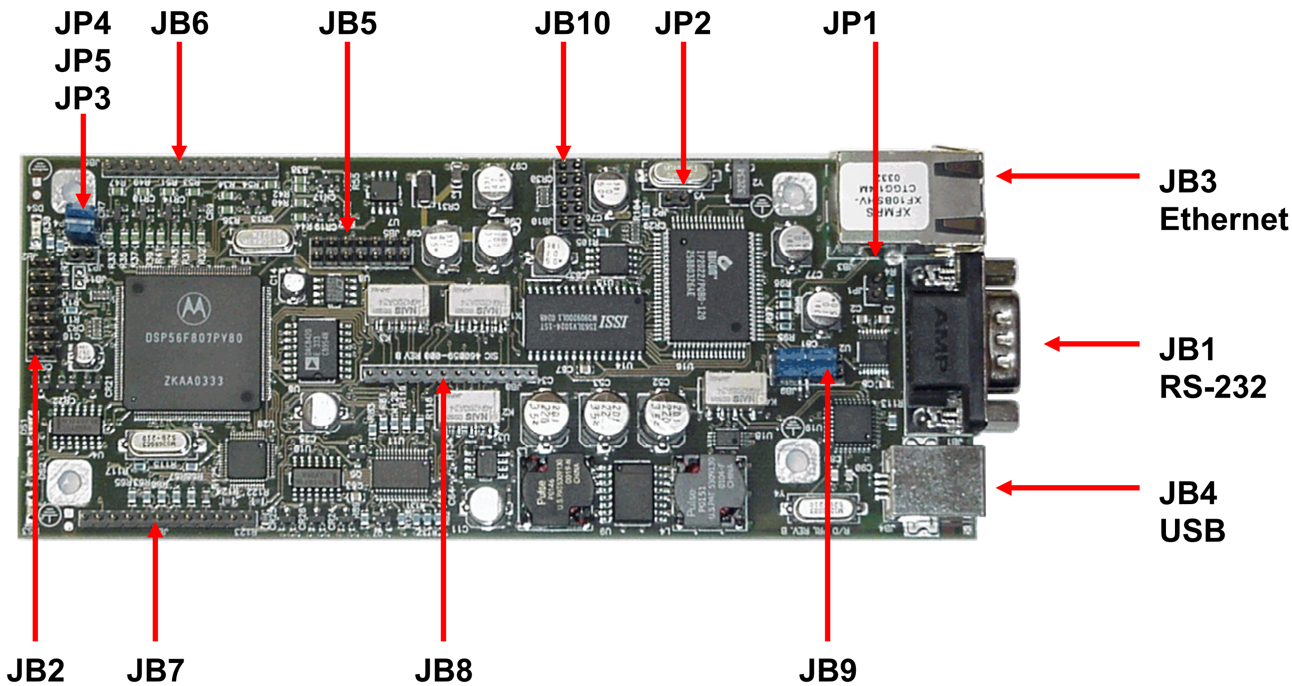


Figure 1B – SIC Rev B assembly

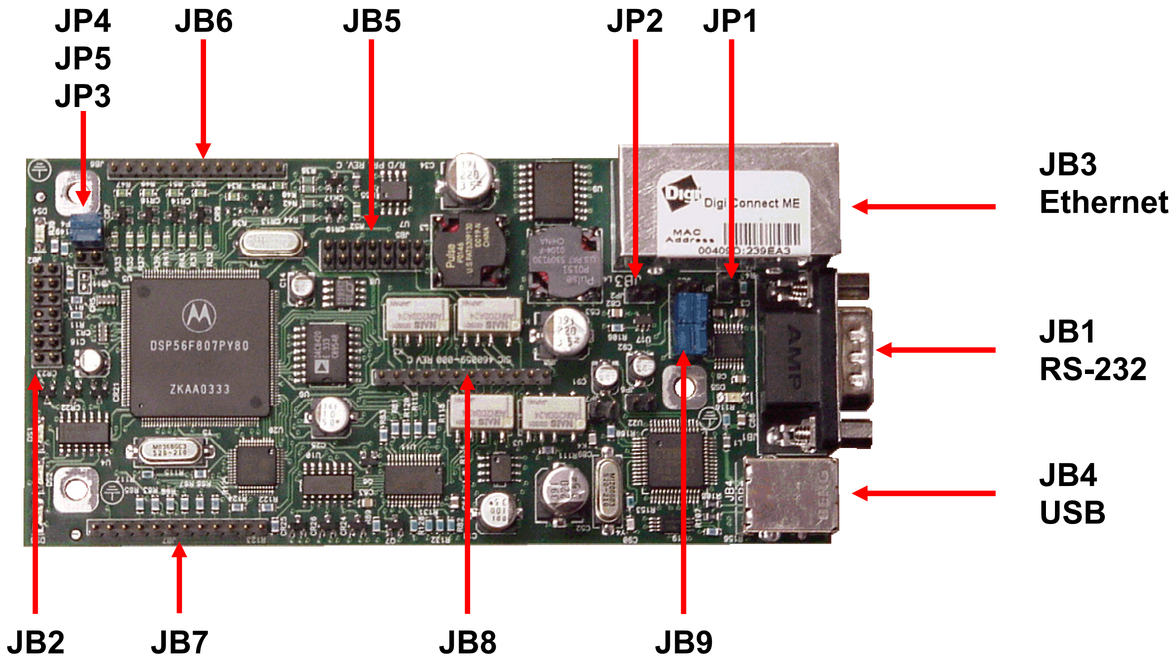


Figure 1C - SIC Rev C Assembly

3.0 GETTING STARTED – HARDWARE SETUP

The digital hardware includes a 40MIPS digital signal processor, a network processor, and a USB processor/controller. Serial port 0 of the DSP is jumper selectable to allow for firmware updating through either the RS-232 port or the Ethernet interface.

3.1 HEADERS / JUMPER BLOCKS

ID	Revision	Description	Normal Connection	Factory Use Only
JB1	All	External RS-232, DB-9	-	-
JB2	All	DSP JTAG	-	Yes
JB3	All	External Ethernet	-	-
JB4	All	External USB, Type B	-	-
JB5	All	Analog, digital, interlock 1	-	-
JB6	All	Analog	-	-
JB7	All	Digital I/O	-	-
JB8	All	Digital I/O, Interlocks 2-3	-	-
JB9	All	Firmware update source	-	-
JB10	A-B	Network JTAG	-	Yes
JP1	All	RS-232 Idle Enable	Open	Yes
JP2	A-B	External Oscillator	Open	Yes
JP2	C & up	DSP-to-Ethernet Reset	Open	Yes
JP3	All	DSP JTAG Enable	Closed	Yes
JP4	B & up	ADC Calibration on ch. 9	Closed	Yes
JP5	B & up	ADC Calibration on ch. 8	Closed	Yes
JP6	C & up	USB Wakeup Select	Open	Yes
JP7	C & up	USB Local F/W Update	Closed	Yes

3.2 CONFIGURING THE HARDWARE

Normally, the end user will not need to configure the SIC board, as it will be pre-configured from the factory for the type of unit it is intended to interface with. However, the user may desire to change the setting of the Firmware Update Source via JB9.

3.2.1 TO UPDATE VIA RS-232:

Set the JB9 jumpers as per Figure 2.
1-3, 2-4, 5-7, 6-8

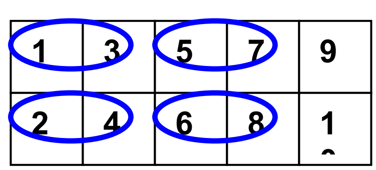


Figure 2 – Firmware Updates via RS-232

3.2.2 TO UPDATE VIA ETHERNET:

Set the JB9 jumpers as per Figure 3.
3-5, 4-6, 7-9, 8-10

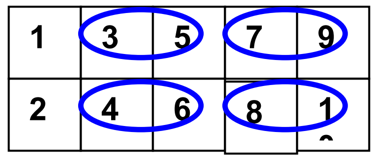


Figure 3 – Firmware Updates via Ethernet

3.3 RS232 INTERFACE

The RS232C interface has the following attributes:

- 115K bits per second
- No Parity
- 8 Data Bits
- 1 Stop Bit
- No handshaking
- DB-9 connector as shown

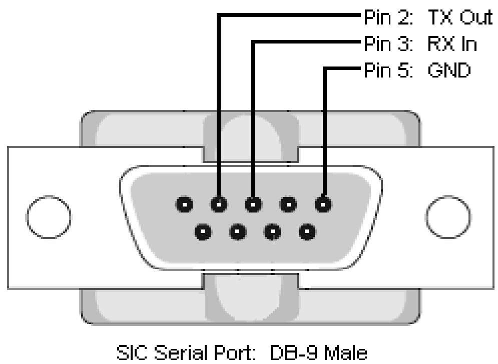


Figure 4 – JB1, RS-232 DB-9M pinout (front view)

PIN	DESCRIPTION
1	-
2	Tx Out
3	Rx In
4	-
5	Ground
6	-
7	-
8	-
9	-

3.4 ETHERNET INTERFACE

The Ethernet interface has the following attributes:

- 10-Base-T (rev A-B), 10/100-Base-T (rev C and higher)
- IP address can be set by the system integrator
- Network Mask can be set by the system integrator
- TCP Port Number can be set by the system integrator
- RJ-45 connector
- Network attachment via Crossover and Standard Ethernet cables.
- Supported Operating Systems: Windows 98 2ED, Windows 2000 (SP2), Windows NT (SP6), Windows XP Professional

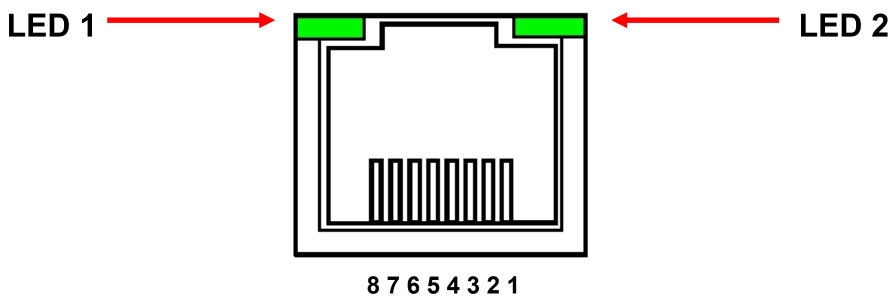


Figure 5 – JB3, Ethernet RJ45 Jack (front view)

PIN	DESCRIPTION
1	TX+
2	TX-
3	RX+
4	-
5	-
6	RX-
7	-
8	-

The Ethernet RJ-45 has two LED indicators, as shown in Figure 5. The left LED, LED1 indicates that the network processor has a valid network link. The right LED, LED2 indicates network activity.

3.5 USB – UNIVERSAL SERIAL BUS INTERFACE

The USB interface has the following attributes:

- Compliant with USB 1.1 and USB 2.0 specifications
- Type B male connector
- Included driver can be communicated with via standard Windows serial communications methods

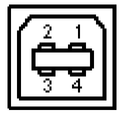


Figure 6 – JB4, USB Type B (front view)

PIN	DESCRIPTION
1	Vbus +5V
2	D-
3	D+
4	Ground

3.6 RS-232 CABLING

A standard RS-232 cable where lines 2 and 3 are reversed is used to connect the SIC serial port to the serial port on a standard personal computer. Please refer to the following chart.

PC to SIC Board Cable Details	
PC Connector (DB-9 Female)	SIC Connector (DB-9 Male)
Pin 2: RX In	Pin 2: TX Out
Pin 3: TX Out	Pin 3: RX In
Pin 5: Ground	Pin 5: Ground

3.7 ETHERNET CABLING

Category 5 (CAT5) Ethernet patch cables are used to connect the SIC to the host computer. There are two ways to connect to the SIC board via Ethernet: the first is to directly cable between the host and the SIC board, and the second is through the use of a switch, hub, or network.

A direct connection requires a non-standard cable where the wires are not run straight through. Please refer to the two cable ends shown below in figure 7.

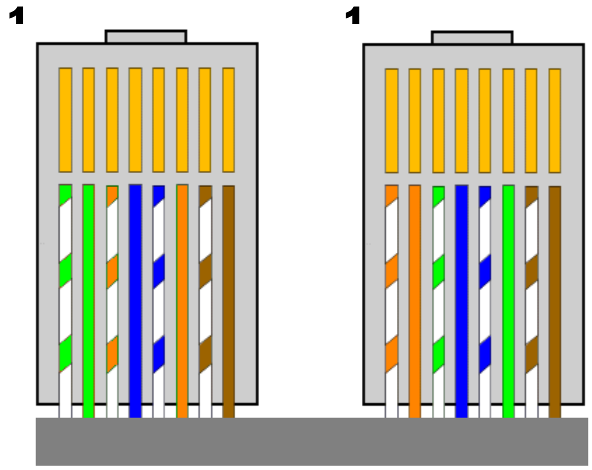


Figure 7 – Crossover Cable for Direct Connection

A standard connection through a hub, switch, or network uses a standard CAT5 patch cable. Please refer to the two cable ends shown below in figure 8.

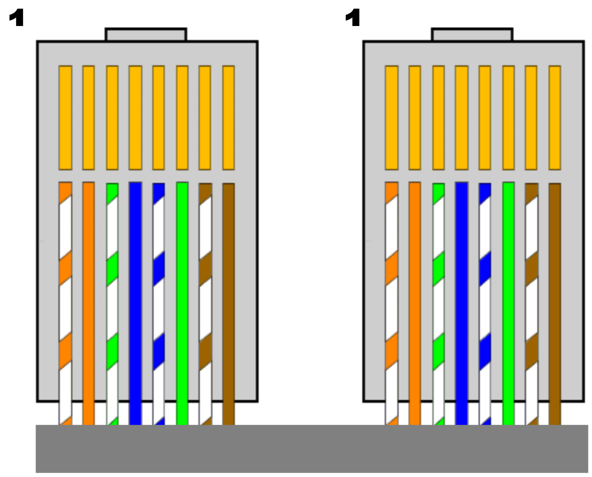


Figure 8 – Standard Straight Through Cable – Standard CAT5 Patch

3.8 USB CABLING

A high-quality double shielded USB 2.0 Type A to B (host to slave) cable should be used in all applications. This type of cable is a standard PC to peripheral cable that utilizes full-size connectors.

Figure 9 – USB A-to-B cable

3.8.1 HIGH EMI ENVIRONMENTS

If the SIC USB interface is being used in a high-EMI environment, ferrites should be added to the USB cable. Figure 10 illustrates the possible combinations of ferrites that can be used to achieve acceptable operation under these conditions.

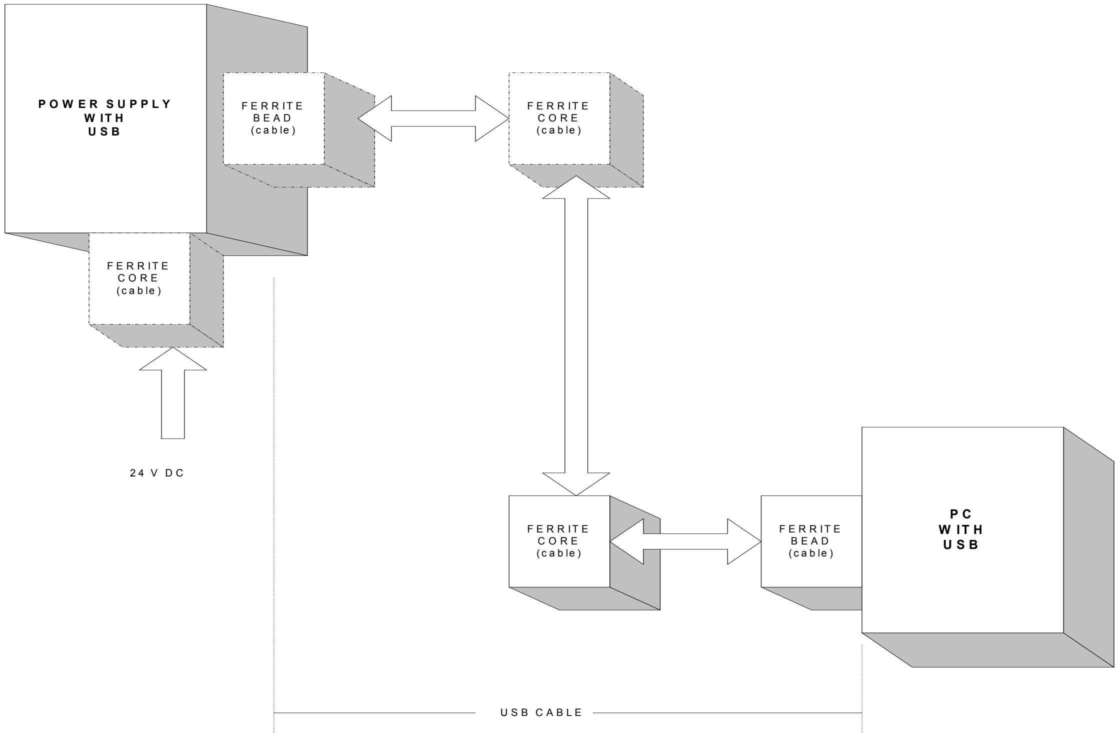


Figure 10 – Block Diagram of USB Cable Utilizing Ferrites

Ferrite beads should be attached to the USB cable next to the connectors – both sides should be installed. In extreme cases ferrite cores may be added where the cable is looped 3 or 4 times around the core as shown in figure 11. Cores of 1.5 to 2 inches should be used at both ends of the cable. In addition, a ferrite core may be required on the 24VDC input.

Figure 11 - Example of a USB Cable Using Ferrites

Please refer to the USB Interface Setup section, for an explanation of how USB works and why EMI may present a problem for this communications interface.

4.0 GETTING STARTED – SOFTWARE

The following sections detail how to create software to interface to the SIC communications interfaces. In addition, please reference section 8 which lists commands used for specific Spellman power supplies.

4.1 RS-232

The RS-232 interface makes use of a standard 'command/response' communications protocol. See section 6.0 for the syntax of the serial interface protocol. The programmer should also review section 4.3 for programming considerations for the USB interface as the code is nearly identical for the RS-232 interface.

All software that addresses the RS-232 interface must adhere to the following parameters:

- 115K bits per second
- No Parity
- 8 Data Bits
- 1 Stop Bit
- No handshaking

4.1.1 Enabling Communications Objects in Visual Basic for RS-232

Communications in Microsoft Visual Basic 6.0 are directed to a control that abstracts the port. In the case of serial and USB we need Microsoft Comm Control 6.0. To enable this in your VB 6 project, go to:

Project -> Components

Then in the list make sure that Microsoft Comm Control 6.0 has a check next to it. The Comm Control Object should then appear in your toolbox. It will have an icon of a telephone and will be named: MSComm. This can be dragged and dropped into your application. You will then need to set the object's properties.

4.1.2 Configuring Communications in Visual Basic for RS-232

In order to configure the MSComm Object, first you must initialize it in the Object properties:

Settings	115200,n,8,1
Handshaking	0 – comNone

The application can be set to either default to a specific COM Port or the End User can be allowed to choose one for the particular PC.

For the “Default” scenario, include the following commands in the Form_Load() routine:

```
MSComm1.CommPort = portNumber
MSComm1.PortOpen = True
```

For the “Choice” scenario, place the above two commands in a selectable menu item.

4.2 ETHERNET

The SIC board contains an embedded diagnostic web server that can be accessed through any standard web browser by browsing to the SIC's IP address. For example:

http://192.168.0.1

The Ethernet interface communicates using the following protocols:

- TCP/IP
- HTTP
- TFTP
- FTP

Revision A and B assemblies can communicate at 10Mb/s, while Revision C and higher assemblies can auto-switch between 10Mb/s and 100Mb/s.

4.2.1 Diagnostic Web Server

The diagnostic web server can control and monitor an SIC equiped power supply from a web browser. It displays operating status of the Power Supply and allows the unit to be configured in real time. The application consists of three web pages; a page displaying contact information, a license agreement, and a monitoring and control applet that is at the heart of this application. The Web Server application for the MNX50P50 is presented as an example in the following pages.

4.2.2 Web Pages

4.2.2.1 Web Page 1: Contact Information Page

Figure 12 displays a picture of the MNX50P50 and information on how to contact Spellman High Voltage Electronics Corporation. By clicking on the picture of the MNX50P50 or on the button labeled "Click Here to Monitor and Control" one can move on to the next screen, the license agreement.

Figure 12 - Web Page 1- Contact Information

4.2.2.2 Web Page 2: License Agreement Page

Figure 13 displays the license agreement. Here the user can either agree or disagree with the Spellman license agreement. Click on “I Accept” to continue on to the applet.

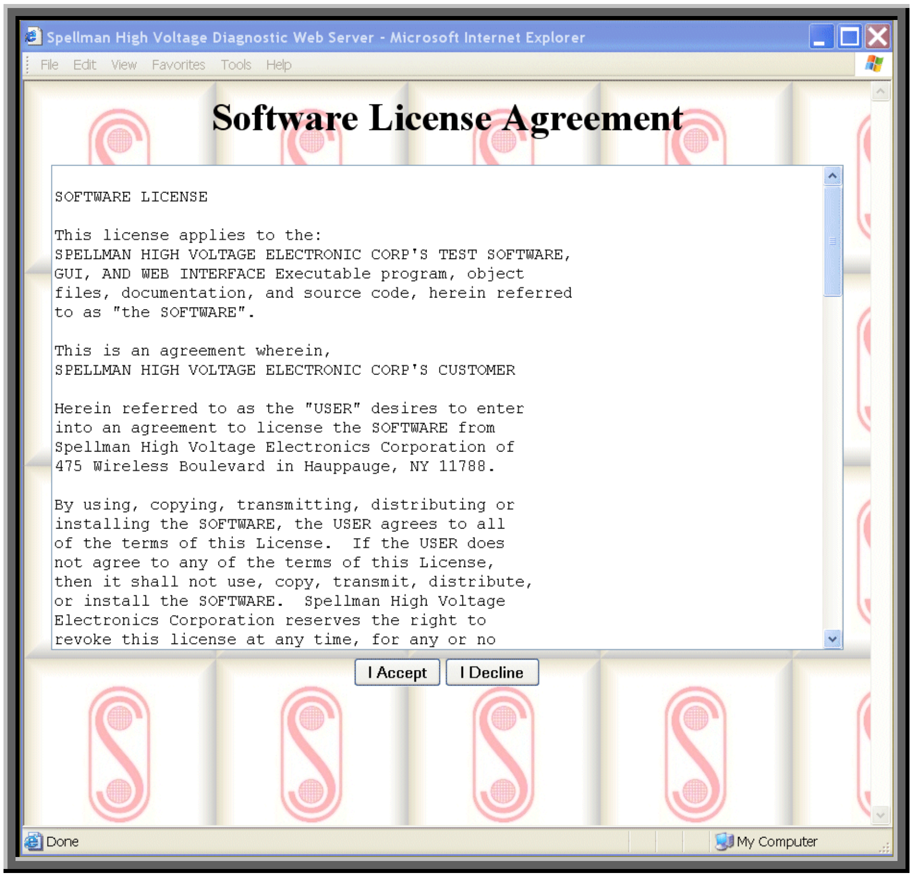


Figure 13 - Web Page 2 – License Agreement

4.2.2.3 Web Page 3 - Monitor and Control Applet

4.2.2.3.1 Requirements

The Monitor and Control Applet is a java "applet" ("small java application" specifically written to be embedded in a web page and invoked from a browser) that requires an Internet browser with an installed JVM (Java Virtual Machine). We have tested under Internet Explorer 5 and 6, Microsoft JVM 5 and Sun JVM versions 1.4.1 and 1.4.2.

4.2.2.3.2 Description of Monitor and Control Applet

Figure 14 displays an example of an embedded monitor and control application.

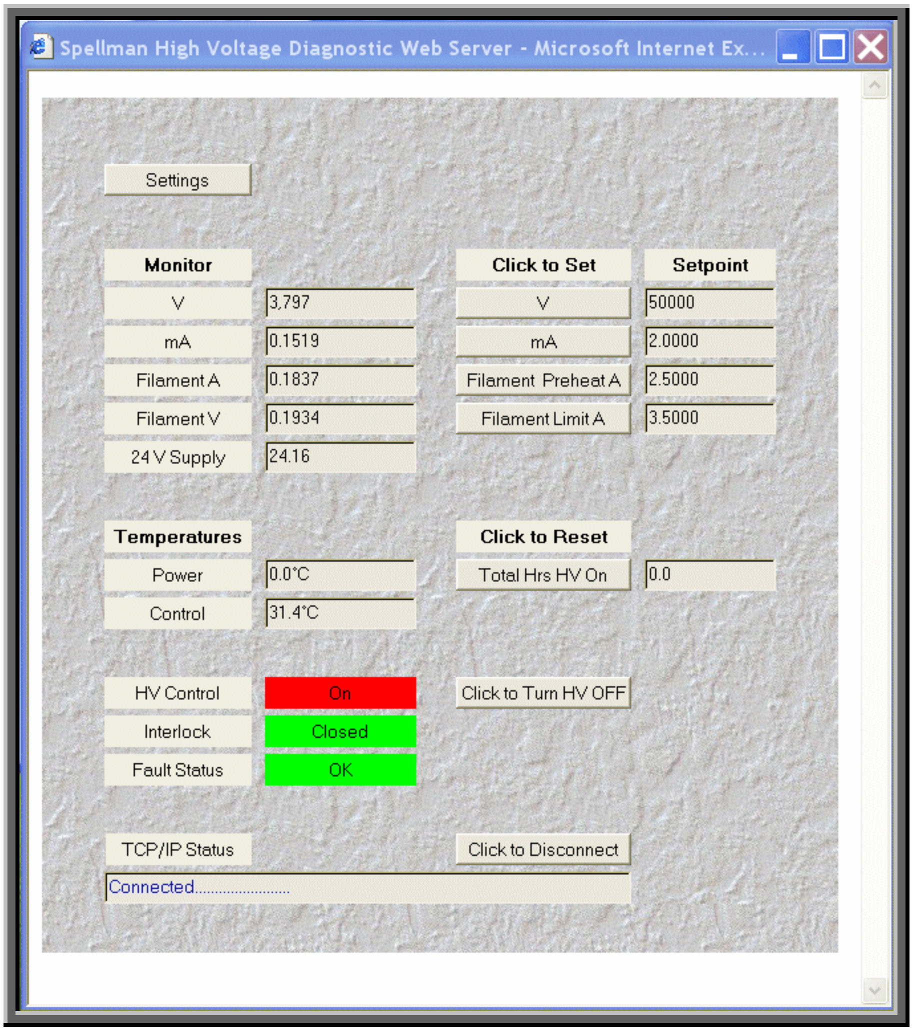


Figure 14 - Control and Monitor Applet

Broadly one can view the screen as a “left” and a “right” with the left half containing status values (read backs) read from the SIC and the right half containing the values that are configurable by the user. Notice that the top of the right half contains the label “Click to Set”. For any configurable setting you click on the button to the left of the setting, which brings up the program set point screen. For example, click on the button labeled, ‘V’ to set the output voltage set point. Refer to figure 15.

4.2.2.4 Program Set Point Screen

On the program set point screen (Reference figure 15) there are two fields: a top field labeld ‘V’ and a bottom field labeled ‘DAC’.

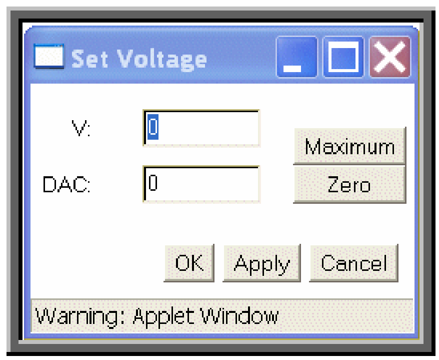


Figure 15 - Program Configurable Values Screen

The top field is the scaled value or real world value, which is the field the user is going to use most of the time. The bottom field is the raw digital-to-analog converter (DAC) value that is actually sent to the SIC. Enter either the desired set point level in the top field or the DAC value between 0 and 4095 in the bottom field.

The user can then click Apply to send the set point to the SIC and remain in the set point screen, or click OK to send the set point and close the set point entry window. The Maximum and Zero buttons will set both fields to the indicated end point value. The user may also click on Cancel to close the window without sending any changes.

4.2.2.5 Java Warning Messages

You may notice a message at the bottom of all dialog windows that are displayed from the SIC Control and Monitor Applet. The wording may vary slightly depending on the JVM version but on some the message is “Warning: Applet Window”. This message is letting you know that the dialog window was generated by an applet. The design philosophy for the JVM was for secure computing so the origins of new windows are supposed to be as obvious as possible.

4.2.2.6 Menu Item "Settings" on Applet

The user can view and set operating parameters of the applet or network configurations of the SIC or view firmware version information for both through the settings menu. Click on the button at the top of the Monitor and Control Applet that has the label "Settings". This displays the settings popup menu as shown in figure 16.

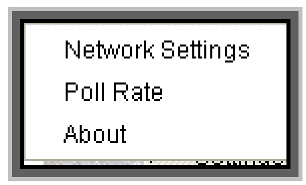


Figure 16 – Settings Pop up Menus

Notice that there are three choices. The first, "Network Settings" refers to the network settings for the network component of the SIC and not the Monitor and Control applet. The second option, "Poll Rate" affects refresh rate of the Monitor and Control Applet and will be discussed in the next section. In the "about" choice firmware version information is displayed, both for the Monitor and Control Applet and for the SIC hardware.

4.2.2.7 Refresh rate for monitored values

The refresh rate for the applet display of the SIC is dependent upon the rate of placement of status requests in the internal send queue and how fast responses are sent back from the SIC in response to the requests. The default value for queuing responses is every 600ms and this is a configurable value in the

Settings->Poll rate screen. Please refer to figure 17.

Figure 17 - Configure Polling Rate Screen

Setting this value lower may make the screen refresh quicker. However, setting it too low may cause requests to queue up in the send queue. This may make controlling the SIC very slow, as control requests now must wait behind queued status requests. We recommend leaving the delay set at the default value.

4.2.2.8 Version Information

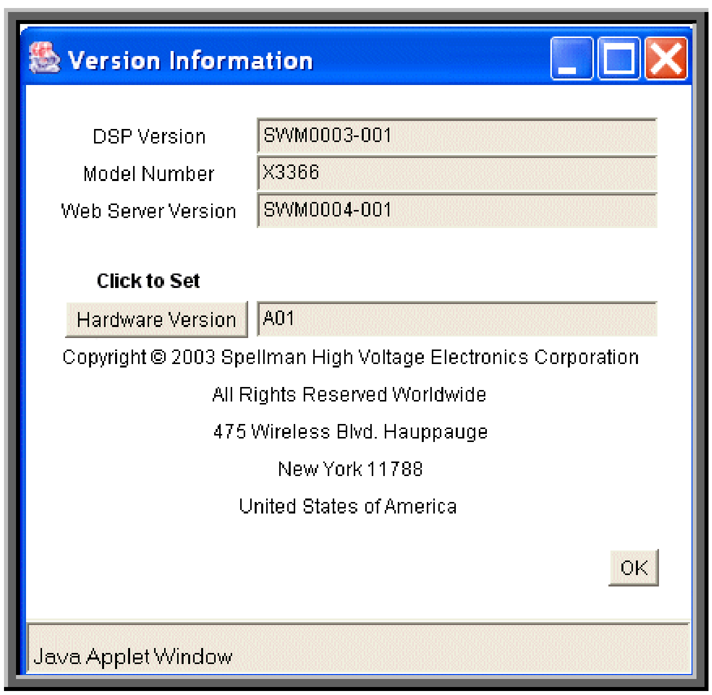


Figure 18 – Version Information

4.2.2.9 Turning the SIC HVOn/Off and Connection Status

Please refer to Figure 14, the Monitor and Control Applet.

Setting Name	Range Values
HV	On/Off
Interlock	Open/Closed
Fault Status	OK/Fault
Connection Status	Connected/No Data Received/Disconnected

Unlike the controls we previously discussed at the top of the screen which required a separate dialog screen to enter values, these are controlled by a button. For example, an On/Off button controls the HV. When HV is on, the Control is labeled “Click to Turn HV Off”. When HV is off, the control is labeled “Click to Turn HV On”. Thereby handling the two distinct states.

Notice that at the very bottom of the screen is a text field that displays the current connection status, which as mentioned above is one of three values. “Connected” is displayed when there exists a valid TCP/IP session connecting the SIC and the Applet and data is being received by the applet from the SIC. The next state is “No Data Received” which is when there is still a valid connection but no responses have been received from the SIC for 2 seconds. Lastly, the text field displays “Disconnected” when the TCP/IP session has been disconnected.

When the Applet is first started and anytime the “Click To Connect” button is clicked there is a 5 second delay as the Applet starts up the threads necessary for communication between it and the SIC.

4.2.2 Direct Connection between the SIC and a Computer

A direct Ethernet connection between the SIC and the computer requires an RJ45 crossover cable. The end connectors will look identical to a “normal” RJ45 connector but the colors of some of the wires in the connectors will be “reversed”. Hold up the two ends of the RJ45 cable and look at the color of the wires from left to right. They should differ on the two connectors.

When direct connecting the SIC to a computer using a crossover cable over Ethernet they are essentially participating in a private network. As such you need to pick two valid IP addresses, one for each device.

The table below illustrates that not all IP addresses are actually valid IP addresses. For example, IP addresses beginning with 127 are not valid.

Class	Address Range
A	1.0.0.0-126.255.255.255
B	128.0.0.0-191.255.255.255
C	192.0.0.0-223.255.255.255

4.2.2.1 Configuring the Computer for Direct Ethernet Connection

As mentioned above both the IP Address and Subnet Mask need to be configured. In our environment computers normally are assigned IP addresses dynamically, using DHCP. We need to change this and assign the IP Address statically to the one we have selected.

Here are the steps on Windows XP. On the desktop right click on "My Network Places" and select properties at the bottom of the menu.

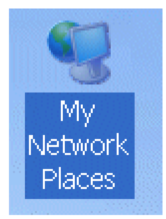


Figure 19 – Right Click on Desktop

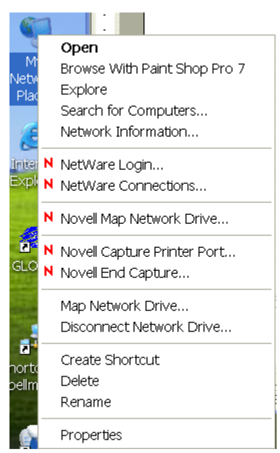


Figure 20 – Select Properties

After selecting properties you are brought up to the screen below (Figure 21). You must RIGHT CLICK and select Properties on Local Area Connection, and not double click which will display a window similar to figure 22.

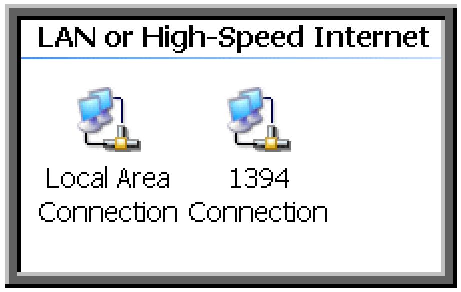


Figure 21 – Here you must Right Click and Select Properties

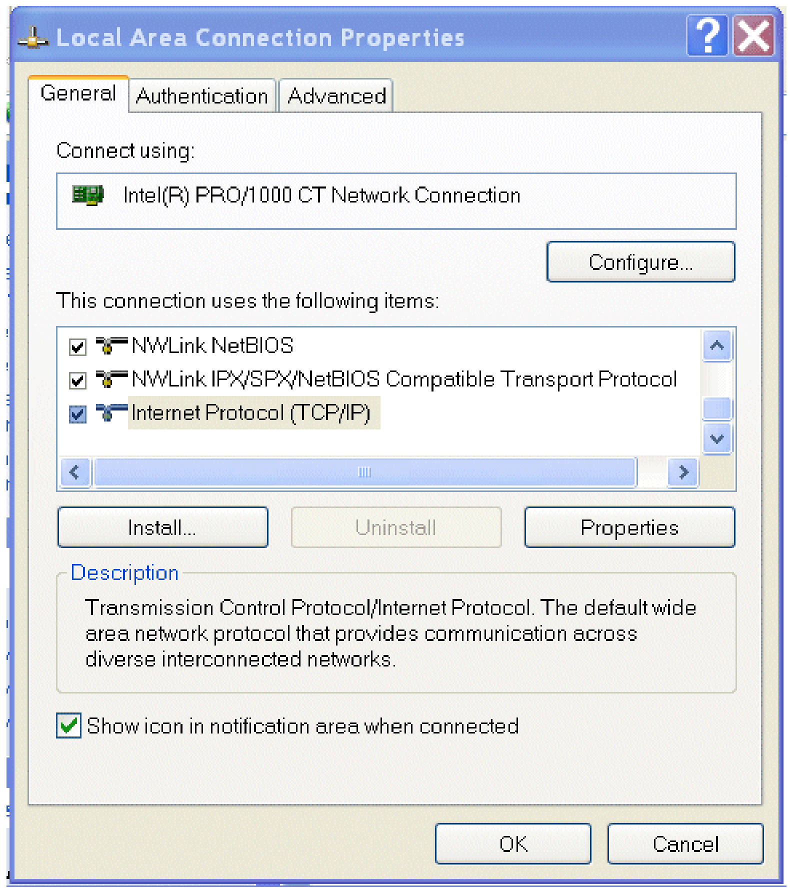


Figure 22 – Local Area Connection Properties

Now you must select "Internet Protocol (TCP/IP)" and click on the Properties button to be brought to figure 23. Lastly you must disable any firewall software you have running. If you are running a proxy server for Internet access, you must also disable the proxy client. Disabling this also requires a reboot.

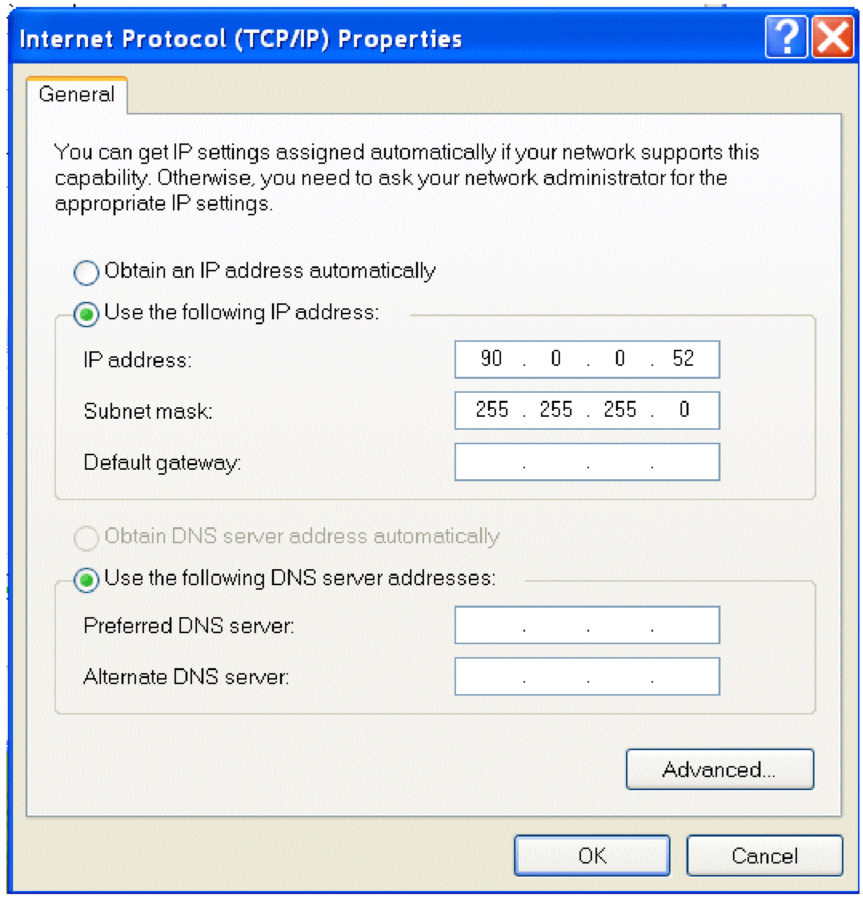


Figure 23 – TCP/IP Properties

4.2.2.2 Testing a Direct Connection

You can use the program “Ping” to test a network connection between the computer and the SIC. “Ping” is a command line tool so we will need to bring up a command prompt. Under Windows NT, 2000 and XP the name of this command is “CMD”. Under Windows 98 the name of this command is “Command”.

To do this, click on Start->Run->Cmd

Then on the command line type

Ping <IP Address>

For example

Ping 192.168.1.2

If the SIC board is found at the specified IP address, the Ping command will respond with a report that is similar to:

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=64

Reply from 192.168.1.2: bytes=32 time<1ms TTL=64

Reply from 192.168.1.2: bytes=32 time<1ms TTL=64

Reply from 192.168.1.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

4.2.3 Configuring the SIC For a Local Area Network (LAN)

If you have chosen to place the SIC onto your local area network you will need:

- A CAT5 network patch cable to physically connect the SIC to the LAN
- A static IP address to assign to the SIC.

Remember that even if the IP address you have selected is in general a valid IP address it needs to be valid for your LAN (local area network). Otherwise the device will not be accessible from an Internet browser or Ping.

4.2.3.1 Configuring the Network Settings from the Monitor and Configure Applet

The network settings are configurable from the Settings->Network Settings screen, refer to figure 24.

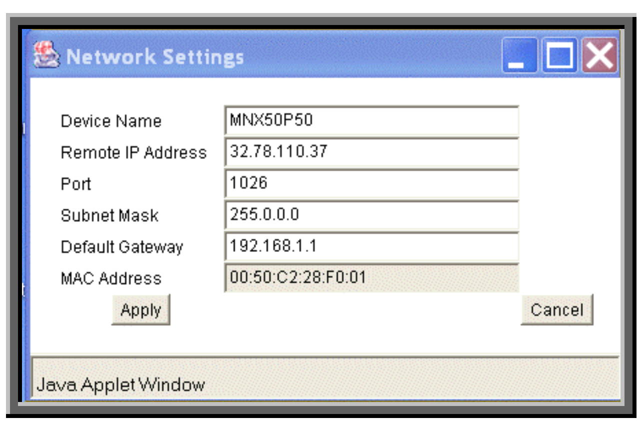


Figure 24 - Configure Network Settings

The settings that can be changed are the:

- Device Name
- IP Address
- TCP Port
- Subnet Mask
- Default Gateway

Once the Apply button is clicked on the network settings screen the network component of the SIC is configured, rebooted and the applet is disconnected from the SIC. You must type the NEW IP address into a web browser to bring up a new instance of the applet to monitor and control the SIC after reconfiguring it. This may also require reconfiguring the host computer with the correct host IP address, subnet mask, and TCP port.

The device name does not affect the operation of the SIC; it is simply a way for the user to differentiate multiple units on the same network.

Depending on the type of network you are attaching the SIC to, you may need to configure the host PC's IP address and subnet mask as shown in section 4.2.2.1. You can also test a network connection to the SIC by following the instructions listed in section 4.2.2.2.

4.2.4 Enabling Communications Objects in Visual Basic for Ethernet Communications

For Ethernet communications, we need Microsoft Winsock Control 6.0 and SP5. To enable this in your VB 6 project, go to:

Project -> Components

Once selected in your toolbox you will have an icon of two computers linked together and it will be named: Winsock.This can be dragged and dropped into your application. Then set the object's properties.

4.2.5 Configuring Communications in Visual Basic for Ethernet

In order to configure the Winsock Object, you must make the following initialization in the object's properties:

Protocol 0 – sckTCPProtocol

Then, in the application code, include the following commands:

```
tcpClient.RemoteHost = host
tcpClient.RemotePort = portNumber
tcpClient.Connect
```

For further information regarding the use of the above commands, please refer to your Visual Studio Help File.

4.2.5.1 Data Output Example

MSComm1 is both the serial and USB port. TcpClient is the Ethernet port.

```
If (portType = "ethernet") Then
        tcpClient.SendData (str)
Else
        MSComm1.InBufferCount = 0
         On Error GoTo done
        MSComm1.Output = str
        done:
        tmrOpenClose.Enabled = True
End If
```

4.2.5.2 Data Input Example

```
If (portType = "ethernet") Then
        Do
                DoEvents
                tcpClient.GetData temp$
                str = str + temp$
                Loop Until InStr(str, Chr(3)) Or Timer - t1 > 1
                On Error Resume Next
Else
        Do
                DoEvents
                If MSComm1.InBufferCount > 0 Then
                        str = str & MSComm1.Input
                End If
                Loop Until InStr(str, Chr(3)) Or Timer - t1 > 1
                If InStr(str, Chr(3)) > 0 Then
                        tmrOpenClose.Enabled = False
                End If
End If
```

4.3 USB

The USB interface makes use of a standard 'command/response' communications protocol. See section 6.0 for the syntax of the serial interface protocol.

The USB interface is accessed through a Windows USB driver that emulates a standard communications port (just like in RS-232). Before you can communicate with the SIC USB interface, you must load the supplied USB driver disk. This driver will create a 'virtual' comm port that can be checked by using Windows Device Manager.

4.3.1 USB Driver Installation

The following steps are valid for Windows 9X or higher:

1. Run the installer named HidComInst.exe for revision C and higher assemblies, or TUSB3410_9x.exe/TUSB3410_2K.exe for revision A-B assemblies. It is best to save the driver files to either a folder on the Desktop or the root directory of the C: drive.

2. Power up the Spellman High Voltage device.

3. Connect a USB cable from the Spellman High Voltage device to an available USB port on the PC.

4. When the "Found New Hardware" dialog asks for the location of the drivers, browse to the directory created in step 1.

5. With the Spellman High Voltage device connected to the monitoring system, open the MY Computer -> Properties page, or go to the Control Panel and click on the System Icon. Open the Device Manager window. In Windows XP systems, this is accessible from the Hardware tab in the System Properties display as shown in Figure 25.

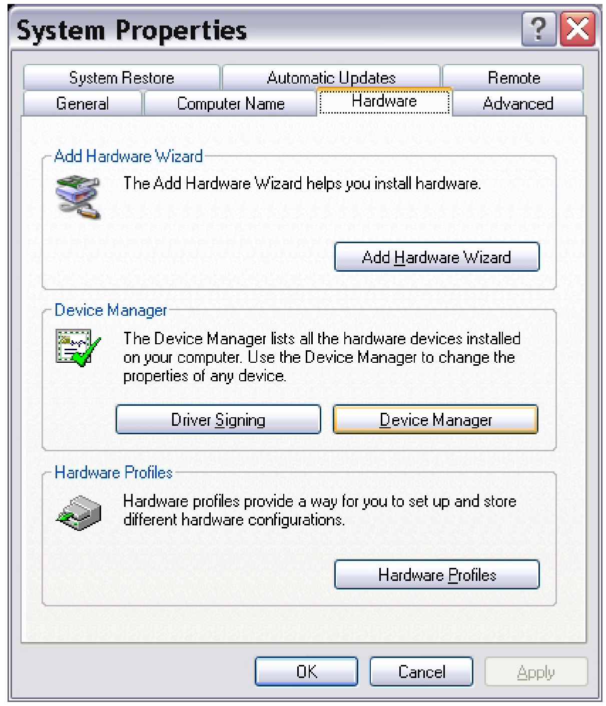


Figure 25 – System Properties

6. Select the "View Devices by Type" display, and click on the icon labeled "Ports (COM & LPT)". Find the Port that is labeled "Cypress USB-HID" for Revision C and higher assemblies, or "TUSB3410" for Revision A-B assemblies. This is the Com Port to which the USB Driver has been assigned.

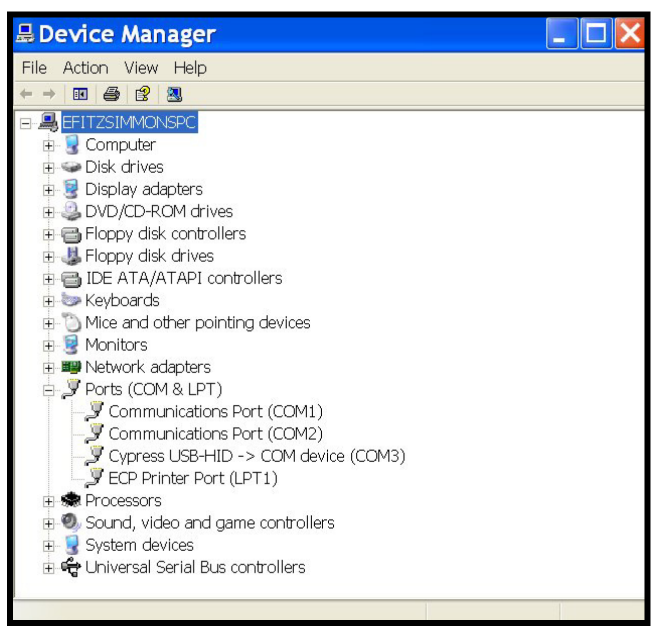


Figure 26 – Device Manager showing USB on COM3

7. Record the Com Port number. This port number will be used in whatever user interface that requires a USB connection. Bear in mind that this Port number is assigned by the Windows operating system and as such, may not be the same number from session to session.

4.3.2 USB and EMI

The USB protocol utilizes a heartbeat signal from each client device back to the host (PC). If the heartbeat is interrupted due to radiated or conducted transient noise, it is possible that the host may lose connection with the client. This can cause problems with data transfers over the USB cable.

The SIC revision C or higher assembly when used in combination with the HIDCOM Windows driver makes it possible for the host to renumerate the client connection and reestablish communications. This is providing the control application implements a method of timeout and retry.

4.3.3 Enabling Communications Objects in Visual Basic for USB

Communications in Microsoft Visual Basic 6.0 are directed to a control that abstracts the port. In the case of serial and USB we need Microsoft Comm Control 6.0. To enable this in your VB 6 project, go to:

Project -> Components

Then in the list make sure that Microsoft Comm Control 6.0 has a check next to it. The Comm Control Object should then appear in your toolbox. It will have an icon of a telephone and will be named: MSComm. This can be dragged and dropped into your application. You will then need to set the object's properties.

4.3.4 Configuring Communications in Visual Basic for USB

In order to configure the MSComm Object, first you must initialize it in the Object properties:

Settings	115200,n,8,1
Handshaking	0 – comNone

The application can be set to either default to a specific COM Port or the End User can be allowed to choose one for the particular PC. For the "Default" scenario, include the following commands in the Form_Load() routine:

```
MSComm1.CommPort = portNumber
MSComm1.PortOpen = True
```

For the “Choice” scenario, place the above two commands in a selectable menu item.

4.3.5 Software Considerations for USB Reconnection

The following Visual Basic code snippets are presented as a guideline for implementation with revision C and higher assemblies.

4.3.5.1 Recognize partial, corrupt, or absent data

```
1:  temp2$ = inputInputString
2:  If temp2$ <> "" Then
3:      btn_UPDATEDATA.Value = False
4:      CommStatusFlag = True
5:      CommaPos = InStr(Start, temp2$, Comma, vbTextCompare)
6:      ' Channel 0
7:      On Error GoTo endhere
8:      AmbTemp = Mid(temp2$, Start, (CommaPos - Start))
```

Please note that even though we have guarded against no data, in line 2, we still need to guard against bad data, in this case no comma, on line 8. If there is no comma, we wind up passing a negative value to Mid, which is an error, that we should trap for.

4.3.5.2 Retrieve data only if it exists

```
1:  Do
2:      DoEvents
3:      If MSComm1.InBufferCount > 0 Then
4:          str = str & MSComm1.Input
5:      End If
6:      Loop Until InStr(str, Chr(3)) Or Timer - t1 > 1
7:      'str = str & MSComm1.Input
8:      If InStr(str, Chr(3)) > 0 Then
9:          tmrOpenClose.Enabled = False
10:     End If
```

Notice that in line 3 we check for the existence of data before we extract data from the USB port. Normally, if there is no data, line 4 would append an empty string. However, during a noise event, retrieving data without first checking the existence of data could hang.

4.3.5.3 Shutdown Communications Port if no data is received

Once a transient noise event occurs, we may need to open and close the port if no further data is being retrieved. So in our output function we start a timer:

```
1:      MSComm1.InBufferCount = 0
2:      On Error GoTo done
3:      MSComm1.Output = str
4:      done:
5:      tmrOpenClose.Enabled = True
6:  End If
```

4.3.5.4 Periodically reconnect to port to test the connection

Our timer (tmrOpenClose) is set for 3 seconds. Every time we send data we enable the timer. In the following function we close and open the port

```
1:  Private Sub tmrOpenClose_Timer()
2:      If MSComm1.PortOpen = True Then
3:              MSComm1.PortOpen = False
4:              On Error GoTo done
5:              MSComm1.PortOpen = True
6:              done:
7:              tmrOpenClose.Enabled = False
8:      End If
9:  End Sub
```

Lastly we have a timer that periodically turns the port on if it is off.

```
1:  If CommStatusFlag = True Then
2:      If MSComm1.PortOpen = False Then
3:              On Error GoTo done
4:              MSComm1.PortOpen = True
5:              done:
6:      End If
```

4.3.5.5 Example Output Routine

Notice that on line 11 we register an error handler in case the port is invalid because we have closed it in another routine. Notice that on line 14 we start a timer. When we output data on the port we start a timer to keep track of incoming data. If we get no incoming data it means that communications have been interrupted.

```
Private Sub outputOutputString(outputString As String)
  Dim str As String
  str = ProcessOutputString(outputString)
   StatusBar1.Panels(4).Text = "TX: " & str
   'StatusBar1.Panels(3).Text = "RX: Waiting"
   If (portType = "ethernet") Then
    tcpClient.SendData (str)
   Else
       MSComm1.InBufferCount = 0

  On Error GoTo done
       MSComm1.Output = str
done:
      tmrOpenClose.Enabled = True
   End If
End Sub
```

4.3.5.6 Example Input Routine

Notice on line 18 we check for data first before extracting data from the input. Then if we have actual data we turn off the timer. Otherwise the timer routine toggles the port open/close.

```
1: Private Function inputInputString() As String
2:    Dim str As String
3:    Dim t1 As Single
4:   Dim temp$
5:    t1 = Timer
6:
7:   If (portType = "ethernet") Then
8:       Do
10:          DoEvents
11:          tcpClient.GetData temp$
12:          str = str + temp$
13:      Loop Until InStr(str, Chr(3)) Or Timer - t1 > 1
14:      On Error Resume Next
15:   Else
16:      Do
17:          DoEvents
18:         If MSComm1.InBufferCount > 0 Then
19:             str = str & MSComm1.Input
20:         End If
21:       Loop Until InStr(str, Chr(3)) Or Timer - t1 > 1
22:
23:       If InStr(str, Chr(3)) > 0 Then
24:          tmrOpenClose.Enabled = False
25:      End If
```

```

      frm_EXTRAS.txt_MSCOMMBUFF.Text = str
      tmr_COMMWDT.Enabled = True
      On Error Resume Next
   End If
   StatusBar1.Panels(3).Text = "RX: " & str
  inputInputString = str
   tmr_RCVTIMER.Enabled = True
End Function
```

4.3.5.7 Example Timer Routine: Toggle Port State

This is the timer routine in which the open/closed state of the port is toggled. If communications are interrupted, the USB device will re-register itself with the OS (vendor term: renumeration). Once this happens, re-opening the port will enable communications. Until the re-registration happens, open operations will fail. Notice line 5 where we register an error handler.

```
1:Private Sub tmrOpenClose_Timer()
2:   If MSComm1.PortOpen = True Then
3:
4:       MSComm1.PortOpen = False
5:       On Error GoTo done
6:      MSComm1.PortOpen = True
7: done:
8:      tmrOpenClose.Enabled = False
9:   End If
10:
11: End Sub
```

4.3.5.8 Example Timer Routine: Port Reconnection

This is another timer routine whose purpose is to turn the port on if it is off. Notice that in line 8 an error handler is called because if the device has not re-registered itself with the OS, an error will be raised.

```
1: Private Sub tmr_COMMWDT_Timer()
2:
3: tmr_COMMWDT.Enabled = False
4:
5: If CommStatusFlag = True Then
6:
7:    If MSComm1.PortOpen = False Then
8:       On Error GoTo done
9:       MSComm1.PortOpen = True
10: done:
```

```
  End If

ElseIf CommStatusFlag = False Then

   If MSComm1.PortOpen = False Then

       MSComm1.PortOpen = True
   Else
       MSComm1.PortOpen = False
   End If

End If
```

4.3.5.9 Data Parsing Example

Here we have an example of a code that parses incoming data. Notice that it makes use of our generic input and output routines. The important consideration is to gracefully handle corrupted input data after a noise event. In this case we may get data, so a test against empty string returns false, but we may not get commas in the correct place. Notice that we register an error handler on line 26 so that the mid function, which would raise an error when given a negative number, is handled.

```
1: Private Sub btn_EMI_Click()
2:   Dim temp2$
3:    Dim Response1$
4:    Dim Response2$
5:    Dim number$
6:    Dim Comma
7:    Dim CommaPos
8:    Dim Start
9:    Dim ODATA$
10:
11: Comma = ","
12: Start = 5
13:
14: If tmr_RCVTIMER.Enabled = True Then
tmr_RCVTIMER.Enabled = False
15: If tmr_NETRCVTMR.Enabled = True Then
tmr_NETRCVTMR.Enabled = False
16:
17: If AutoUpdate = True Then
18:    tmr_UPDATE.Enabled = False
19: End If
20:
21:    number$ = "15,"
```

```
  outputOutputString (number$)

  temp2$ = inputInputString
  CommaPos = InStr(Start, temp2$, Comma, vbTextCompare)
   On Error GoTo endhere
  Response1$ = Mid(temp2$, Start, (CommaPos - Start))

  'With a 5v reference:
  ODATA$ = Format(str(Response1$ * 0.0004884), "0.##0")

 txt_DACB.Text = ODATA$ + " mA"
  frm_RAWDATA.txt_RAWDACB.Text = str(Response1$)
  txt_DACB.BackColor = vbWhite
  CommStatusFlag = True
endhere:

If portType = "ethernet" Then
 tmr_NETRCVTMR.Enabled = True
Else
  tmr_RCVTIMER.Enabled = True
End If

If AutoUpdate = True Then tmr_UPDATE.Enabled = True
End Sub
```

5.0 ETHERNET COMMANDS

5.1 TCP/IP FORMAT

Each Ethernet command will consist of a TCP/IP header followed by the required data bytes. Figure 27 summarizes the TCP/IP header configuration. Please note that this functionality is provided by the software implementation of the Open Systems Interconnection (OSI) TCP/IP protocol stack, specifically the upper 4 layers.

Byte				
0	Protocol Version	Header Length	Type Of Service	Total Length
4	Packet ID		Flags	Fragmentation Offset
8	Time To Live		Protocol	Header checksum
12	Source Address			
16	Destination Address			
20	Source Port		Destination Port	
24	Sequence Number			
28	Acknowledgement Number			
32	Data Offset	Reserved	Code Bits	Window
36	Checksum		Urgent Pointer	
40	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte N

Figure 27: Network TCP/IP datagram header

The format of Data Bytes 1 through N are as follows:

<STX><CMD><,>ARG><,><ETX>

Where:

<STX>	= 1 ASCII 0x02 Start of Text character
<CMD>	= 2 ASCII characters representing the command ID
<,>	= 1 ASCII 0x2C character
<ARG>	= Command Argument
<,>	= 1 ASCII 0x2C character
<ETX>	= 1 ASCII 0x03 End of Text character

5.2 COMMAND ARGUMENTS

The format of the numbers is a variable length string. To represent the number 42, the string ‘42’, ‘042’, or ‘0042’ can be used. This being the case, commands and responses that carry data are variable in length.

5.3 COMMAND OVERVIEW

Data Byte section of the TCP/IP Datagram			
Command Name	<CMD>	<ARG>	RANGE
Program DAC Channel A	10	1-4 ASCII	0-4095
Program DAC Channel B	11	1-4 ASCII	0-4095
Program DAC Channel D	12	1-4 ASCII	0-4095
Program DAC Channel C	13	1-4 ASCII	0-4095
Request DAC A Setpoint	14	None	-
Request DAC B Setpoint	15	None	-
Request DAC D Setpoint	16	None	-
Request DAC C Setpoint	17	None	-
Request Analog Readbacks – J6 Channels 7 - 15	19	None	-
Request Analog Readbacks – J5 Channels 0 - 6	20	None	-
Request HV On Hours Counter	21	None	-

Request Status	22	None	-
Request Software Version	23	None	-
Request Hardware Version	24	None	-
Request Web Server Version	25	None	-
Request Model Number	26	None	-
Reset HV On Hours Counter	30	None	-
Reset Faults	31	None	-
Request Network Settings	50	None	-
Program Network Settings	51	6 ASCII	See Description
Program Interlock 1	52	1 ASCII	0 or 1
Program Interlock 2	53	1 ASCII	0 or 1
Program Interlock 3	54	1 ASCII	0 or 1
Read Interlock Status	55	None	-
Readback A/D Channel 0 Data	60	None	-
Readback A/D Channel 1 Data	61	None	-
Readback A/D Channel 2 Data	62	None	-
Readback A/D Channel 3 Data	63	None	-
Readback A/D Channel 4 Data	64	None	-
Readback A/D Channel 5 Data	65	None	-
Readback A/D Channel 6 Data	66	None	-
Readback A/D Channel 7 Data	67	None	-
Readback A/D Channel 8 Data	68	None	-
Readback A/D Channel 9 Data	69	None	-

Readback A/D Channel 10 Data	70	None	-
Readback A/D Channel 11 Data	71	None	-
Readback A/D Channel 12 Data	72	None	-
Readback A/D Channel 13 Data	73	None	-
Readback A/D Channel 14 Data	74	None	-
Readback A/D Channel 15 Data	75	None	-
Read Digital Inputs	76	None	-
Program Digital Output Channel 1	84	1 ASCII	0 or 1
Program Digital Output Channel 2	85	1 ASCII	0 or 1
Program Digital Output Channel 3	86	1 ASCII	0 or 1
Program Digital Output Channel 4	87	1 ASCII	0 or 1
Program Digital Output Channel 5	88	1 ASCII	0 or 1
Toggle Verbose Mode	92	None	-
Program High Voltage Status	99	1 ASCII	0 or 1

5.4 RESPONSE OVERVIEW

The command responses will follow the same network TCP/IP header format as outlined above in section 5.1. This list is comprised of Commands with complex responses only. Commands using a simple response will use the <$> character (ASCII 0x24) as a "Success" response or a single character error code. These will be seven ASCII characters in length.

Response Name	<CMD>	Response
Request DAC A Setpoint	14	10 ASCII
Request DAC B Setpoint	15	10 ASCII
Request DAC D Setpoint	16	10 ASCII
Request DAC C Setpoint	17	10 ASCII
Request Analog Readbacks – J6	19	23-50 ASCII
Request Analog Readbacks – J5	20	19-40 ASCII
Request Total Hours High Voltage On	21	13 ASCII
Request Status	22	11 ASCII
Request DSP Software Version	23	17 ASCII
Request Hardware Version	24	9 ASCII
Request Web Server Version	25	17 ASCII
Request Model number	26	11 ASCII
Request Network Settings	50	48-104 ASCII
Read Interlock Status	55	11 ASCII
Readback A/D Channel 0 Data	60	7-10 ASCII
Readback A/D Channel 1 Data	61	7-10 ASCII
Readback A/D Channel 2 Data	62	7-10 ASCII

Readback A/D Channel 3 Data	63	7-10 ASCII
Readback A/D Channel 4 Data	64	7-10 ASCII
Readback A/D Channel 5 Data	65	7-10 ASCII
Readback A/D Channel 6 Data	66	7-10 ASCII
Readback A/D Channel 7 Data	67	7-10 ASCII
Readback A/D Channel 8 Data	68	7-10 ASCII
Readback A/D Channel 9 Data	69	7-10 ASCII
Readback A/D Channel 10 Data	70	7-10 ASCII
Readback A/D Channel 11 Data	71	7-10 ASCII
Readback A/D Channel 12 Data	72	7-10 ASCII
Readback A/D Channel 13 Data	73	7-10 ASCII
Readback A/D Channel 14 Data	74	7-10 ASCII
Readback A/D Channel 15 Data	75	7-10 ASCII
Read Digital Inputs	76	8 ASCII
Read Digital Output Settings	89	8 ASCII
Program High Voltage Status	99	7 ASCII

5.5 COMMAND STRUCTURE

5.5.1 Program DAC Channel A

Description:
The host requests that the firmware change the setpoint of DAC Channel A.

Direction:
Host to supply

Syntax:
<STX><10><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>10,4095,<ETX>

Response:
<STX><10><,><$><,><ETX>
<STX><10><,><ARG><,><ETX>

where <ARG> = error code

Error Codes TBD, 1 = out of range

5.5.2 Program DAC Channel B

Description:
The host requests that the firmware change the setpoint of DAC Channel B.

Direction:
Host to supply

Syntax:
<STX><11><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>11,4095,<ETX>

Response:
<STX><11><,><$><,><ETX>
<STX><11><,><ARG><,><ETX>

where <ARG> = error code

Error Codes TBD, 1 = out of range

5.5.3 Program DAC Channel D

Description:
The host requests that the firmware change the setpoint of DAC Channel D.

Direction:
Host to supply

Syntax:
<STX><12><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>12,4095,<ETX>

Response:
<STX><12><,><$><,><ETX>
<STX><12><,><ARG><,><ETX>

where <ARG> = error code

Error Codes TBD, 1 = out of range

5.5.4 Program DAC Channel C

Description:
The host requests that the firmware change the setpoint of DAC Channel C.

Direction:
Host to supply

Syntax:
<STX><13><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>13,4095,<ETX>

Response:
<STX><13><,><$><,><ETX>
<STX><13><,><ARG><,><ETX>

where <ARG> = error code

Error Codes TBD, 1 = out of range

5.5.5 Request DAC A Setpoint

Description:
The host requests that the firmware report the DAC Channel A setpoint.

Direction:
Host to supply

Syntax:
<STX><14><,><ETX>

Response:
<STX><14><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>14,4095,<ETX>

5.5.6 Request DAC B Setpoint

Description:
The host requests that the firmware report the current DAC Channel B setpoint.

Direction:
Host to supply

Syntax:
<STX><15><,><ETX>

Response:
<STX><15><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>15,4095,<ETX>

5.5.7 Request DAC D Setpoint

Description:
The host requests that the firmware report the current DAC Channel D setpoint.

Direction:
Host to supply

Syntax:
<STX><16><,><ETX>

Response:
<STX><16><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>16,4095,<ETX>

5.5.8 Request DAC C Setpoint

Description:
The host requests that the firmware report the current DAC Channel C setpoint.

Direction:
Host to supply

Syntax:
<STX><17><,><ETX>

Response:
<STX><17><,><ARG><,><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>17,4095,<ETX>

5.5.9 Request Analog Readbacks – J6

<u>Description:</u>
The host requests that the firmware transmit the present values of Analog Channels 7 through 15, which are available via connector J6.

<u>Direction:</u>
Host to supply

<u>Syntax:</u>
<STX><19><,><ETX>

<u>Example:</u>
<STX><19>,<ETX>

<u>Response:</u>
<STX><19><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,>
<ARG5><,><ARG6><,><ARG7><,><ARG8><,><ARG9><,><ETX>

Where:
ARGx = 0 - 4095

<u>Example:</u>
<STX><19>,4095,4095,4095,4095,4095,4095,4095,4095,4095,<ETX>

5.5.10 Request Analog Readbacks – J5

<u>Description:</u>
The host requests that the firmware transmit the present values of Analog Channels 0 through 6, which are available via connector J5.

<u>Direction:</u>
Host to supply

<u>Syntax:</u>
<STX><20><,><ETX>

<u>Example:</u>
<STX>20,<ETX>

<u>Response:</u>
<STX><20><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,>
<ARG5><,><ARG6><,><ARG7><,><ETX>

Where:
ARGx = 0 - 4095

<u>Example:</u>
<STX>20,4095,4095,4095,4095,4095,4095,4095,<ETX>

5.5.11 Request Total Hours High Voltage On

Description:
The host requests that the firmware sends the present value of the Total Hours High Voltage On.

Direction:
Host to supply

Syntax:
<STX><21><,><ETX>

Example:
<STX>21,<ETX>

Response:
<STX><21><,><ARG1>< ARG2>< ARG3><ARG4><ARG5>
<.><ARG6><,><ETX>

Where:
<.> = ASCII 0x2E
ARGx =0-9 in ASCII format

Example:
<STX>21,99999.9,<ETX>

5.5.12 Request Status

Description:
The host requests that the firmware sends the power supply status.

Direction:
Host to supply

Syntax:
<STX><22><,><ETX>

Example:
<STX>22,<ETX>

Response:
<STX><22><,><ARG1><,><ARG2><,><ARG3><,><ETX>

Where:
<ARG1> 1 = HvOn, 0 = HvOff
<ARG2> 1 = Interlock 1 Open, 0 = Interlock 1 Closed
<ARG3> 1 = Fault Condition, 0 = No Fault

Example:
<STX>22,1,1,0,<ETX>

NOTE: This response will also be sent in an unsolicited manner when a change of state is detected on the HvOn and Interlock 1 bits. This is providing that a valid handle has already been established with a host.

5.5.13 Request DSP Software Part Number/Version

Description:
The host requests that the firmware sends the DSP firmware version.

Direction:
Host to supply

Syntax:
<STX><23><,><ETX>

Example:
<STX>23,<STX>

Response:
<STX><23><,>< ARG><,><ETX>

Where:
<ARG> consists of eleven ASCII characters representing the current firmware part number/version. The format is SWM9999-999

Example:
<STX>23,SWM9999-999,<ETX>

5.5.14 Request Hardware Version

Description:
The host requests that the firmware sends the hardware version.

Direction:
Host to supply

Syntax:
<STX><24><,><ETX>

Example:
<STX>24,<ETX>

Response:
<STX><24><,>< ARG><,><ETX>

Where:
<ARG> consists of 3 ASCII characters representing the hardware version. The format is ANN, where A is an alpha character and N is a numeric character

Example:
<STX>24,A01,<ETX>

5.5.15 Request Webserver Software Part Number/Version

Description:
The host requests that the firmware sends the Web Server firmware part number/version.

Direction:
Host to supply

Syntax:
<STX><25><,><ETX>

Example:
<STX>25,<ETX>

Response:
<STX><25><,><ARG><,><ETX>

Where:
<ARG> consists of eleven ASCII characters representing the current firmware part number/version. The format is SWM9999-999

Example:
<STX>25,SWM9999-999,<ETX>

5.5.16 Request Model Number

Description:
The host requests that the firmware sends the unit model number

Direction:
Host to supply

Syntax:
<STX><26><,><ETX>

Example:
<STX>26,<ETX>

Response:
<STX><26><,><ARG><,><ETX>

Where:
<ARG> consists of five ASCII characters representing the model number.
The format is XNNNN, where N is a numeric character.

Example:
<STX>25,X9999,<ETX>

5.5.17 Reset Run Hours

Description:
The host requests that the firmware resets the run hour counter.

Direction:
Host to supply

Syntax:
<STX><30><,><ETX>

Example:
<STX>30,<ETX>

Response:
<STX><30><,><$><,><ETX>

5.5.18 Reset Faults

Description:
The host requests that the firmware resets all Fault messages and indicators.

Direction:
Host to supply

Syntax:
<STX><31><,><ETX>

Example:
<STX>31,<ETX>

Response:
<STX><31><,><$><,><ETX>

5.5.19 Request Network Settings

Description:
The host requests that the firmware transmits the network settings

Application:

	ARG 1	ARG2	ARG3	ARG4	ARG5	ARG6
Function	Device Name	Remote Address	Remote Port	Subnet Mask	Default Gateway	MAC Address

rection:
Host to supply

Syntax:
<STX><50><,><ETX>

Example:
<STX>50,<ETX>

Response:
<STX><50><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,>
<ARG5><,><ARG6><,><ETX>

Arguments:
Device Name is limited to 20 characters or less. Remote address is a ip address in dotted notation. Remote port is a decimal number. Subnet Mask and Default Gateway are also dotted notation and MAC address is in MAC Address notation.

ARG1:	Device Name	1 character minimum, up to 20 maximum
ARG2:	IP Address	<nnn><.><nnn><.><nnn><.><nnn>, where <nnn> represents a number from 0 to 255.
ARG3:	Remote Port	5001 or from 49152 to 65535.
ARG4:	Subnet Mask	<xxx><.><xxx><.><xxx><.><xxx>, where <xxx> represents a number from 0 to 255.
ARG5:	Default Gateway	<yyy><.><yyy><.><yyy><.><yyy>, where <yyy> represents a number from 0 to 255.
ARG6:	MACAddress	<zzz><:><zzz><:><zzz><:><zzz><:><zzz><:><zzz> , where <zzz> represents a number from 0 to 255.

Example:
<STX>50,Spellman2.0,32.78.110.37,1026,255.0.0.0,192.168.1.1,0:100:33:1:32:84,<ETX>

5.5.20 Program Network Settings

Description:
The host requests that the firmware programs the network settings and then reboots.

Application:

	ARG 1	ARG2	ARG3	ARG4	ARG5	ARG6
Function	Device Name	Remote Address	Remote Port	Subnet Mask	Default Gateway	MAC Address

Direction:
Host to supply

Syntax:
<STX><51><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,>
<ARG5><,><ARG6><,><ETX>

Arguments:
Device Name is limited to 20 characters or less. Remote address is a ip address in dotted notation. Remote port is a decimal number. Subnet Mask and Default Gateway are also dotted notation and MAC address is in MAC Address notation.

ARG1:	Device Name	1 character minimum, up to 20 maximum
ARG2:	IP Address	<nnn><.><nnn><.><nnn><.><nnn>, where <nnn> represents a number from 0 to 255.
ARG3:	Remote Port	5001 or from 49152 to 65535.
ARG4:	Subnet Mask	<xxx><.><xxx><.><xxx><.><xxx>, where <xxx> represents a number from 0 to 255.
ARG5:	Default Gateway	<yyy><.><yyy><.><yyy><.><yyy>, where <yyy> represents a number from 0 to 255.
ARG6:	MACAddress	<zzz><:><zzz><:><zzz><:><zzz><:><zzz><:><zzz> , where <zzz> represents a number from 0 to 255.

Example:
<STX>51,Spellman2.0,32.78.110.37,1026,255.0.0.0,192.168.1.1,0:100:33
:1:32:84,<ETX>

Response:
None, as Embedded server reboots with new settings.

5.5.21 Program Interlock State

Description:
The host requests that the firmware Program the state of a specific Interlock Channel.

Direction:
Host to supply

Syntax:
<STX><CH><,><ARG><,><ETX>
Where CH is the command for a specific Interlock Channel, and ARG is a 1 or a 0 to set or clear the interlock.

Channel Number	Command
Interlock 1:	52
Interlock 2:	53
Interlock 3:	54

Response:
<STX><CH><,><$><,><ETX>

Example:
<STX>52,1,<ETX>
Where 1 signifies that interlock channel 1 has been energized.

5.5.22 Read Interlock Status

Description:
The host requests that the firmware read the status of all interlock channels.

Direction:
Host to supply

Syntax:
<STX><55><,><ETX>

Response:
<STX><55><,><ARG1><,><ARG2><,><ARG3><,><ETX>
Where ARG1 through ARG3 are Interlocks 1 through 3. A 1 indicates that the Interlock is energized

Example:
<STX>55,<ETX>

5.5.23 Readback A/D Channel Data

Description:
The host requests that the firmware report data from a specific Analog Channel.

Direction:
Host to supply

Syntax:
<STX><CH><,><ETX>
Where CH is the command representing a specific A/D Channel:

Channel Number	Command	Channel Number	Command
Channel 0:	60	Channel 8:	68
Channel 1:	61	Channel 9:	69
Channel 2:	62	Channel 10:	70
Channel 3:	63	Channel 11:	71
Channel 4:	64	Channel 12:	72
Channel 5:	65	Channel 13:	73
Channel 6:	66	Channel 14:	74
Channel 7:	67	Channel 15:	75

Response:
<STX><CH><,><ARG><,><ETX>

Where:
<ARG>=0-4095 in ASCII format representing unscaled A/D Channel data.

Example:
<STX>68,4095,<ETX>

Note:
Channel 0 is the Ambient Temperature Monitor and Channel 1 is the S.I.C. Board Power Supply Monitor.

5.5.24 Read Digital Input Status

Description:
The host requests that the firmware report the current status of the digital inputs.

Direction:
Host to supply

Syntax:
<STX><76><,><ETX>

Response:
<STX><76><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,><ARG5>
<,><ARG6><,><ARG7><,><ARG8><,><ETX>

Where:
<ARGn> = ASCII Characters 1 or 0 (0x31 or 0x30) representing Digital Input Channel data. Digital Input Channel 1 is represented by ARG1.

Example:
<STX>76,1,1,1,1,1,1,1,1,<ETX>
Where all input channels are detecting binary 1s.

5.5.25 Program a Digital Output Channel

Description:
The host requests that the firmware SET or CLEAR a Digital Output Channel.

Direction:
Host to supply

Syntax:
<STX><CH><,><ARG><,><ETX>
Where CH is the command for a specific Digital Output Channel, and ARG is a 1 or a 0 to set or clear the output.

Channel Number	Command
Channel 1:	84
Channel 2:	85
Channel 3:	86
Channel 4:	87
Channel 5:	88

Response:
<STX><CH><,><$><,><ETX>

Example:
<STX>86,1,<ETX>
The above command sets output channel 3 to 1.

5.5.26 Read Digital Output Settings

Description:
The host requests that the firmware report the current status of the digital outputs.

Direction:
Host to supply

Syntax:
<STX><89><,><ETX>

Response:
<STX><89><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,><ARG5>
<,><ETX>

Where:
<ARGn> = ASCII Characters 1 or 0 (0x31 or 0x30) representing Digital Input Channel data. Digital Output Channel 1 is represented by ARG1.

Example:
<STX>89,1,1,1,1,1,<ETX>
Where all output channels are set to 1s.

5.5.27 Toggle Verbose Mode

Description:
The host requests that the firmware provide or cease continuous data updates via the current communications channel. Transmit this command to begin the transmission of data, transmit again to stop transmission.

Direction:
Host to supply

Syntax:
<STX><92><,><ETX>

Response:
<STX><92><,><$><,><ETX>

Example:
<STX>92,<ETX>

5.5.28 Program High Voltage On/Off

Description:
The host requests that the firmware to turn on or off High Voltage.

Direction:
Host to supply

Syntax:
<STX><99><,><ARG><,><ETX>

Where:
<ARG> 1 = On, 0 = Off in ASCII format

Example:
<STX>99,1,<ETX>

Response:
<STX><99><,><$><,><ETX>
<STX><99><,><ARG><,><ETX>

where <ARG> = error code

Error Codes TBD,
1 = out of range
2 = Interlock 1 open, High Voltage Disabled

6.0 SERIAL COMMANDS – RS-232 / USB

6.1 SERIAL INTERFACE PROTOCOL

Serial communications will use the following protocol:

<STX><CMD><,>ARG><,><CSUM><ETX>

Where:

<STX>	= 1 ASCII 0x02 Start of Text character
<CMD>	= 2 ASCII characters representing the command ID
<,>	= 1 ASCII 0x2C character
<ARG>	= Command Argument
<,>	= 1 ASCII 0x2C character
<CSUM>	= Checksum (see section 6.3 for details)
<ETX>	= 1 ASCII 0x03 End of Text character

6.2 COMMAND ARGUMENTS

The format of the numbers is a variable length string. To represent the number 42, the string ‘42’, ‘042’, or ‘0042’ can be used. This being the case, commands and responses that carry data are variable in length.

6.3 CHECKSUMS

The checksum is computed as follows:

- Add the <CMD>, <,>, and <ARG> bytes into a 16 bit (or larger) word. The bytes are added as unsigned integers.
- Take the 1’s compliment (negate it).
- Truncate the result down to the eight least significant bits.
- Clear the most significant bit (bit 7) of the resultant byte, (bitwise AND with 0x7F).
- Set the next most significant bit (bit 6) of the resultant byte (bitwise OR with 0x40).

Using this method, the checksum is always a number between 0x40 and 0x7F. The checksum can never be confused with the <STX> or <ETX> control characters, since these have non-overlapping ASCII values.

If the DSP detects a checksum error, the received message is ignored – no acknowledge or data is sent back to the host. A timeout will act as an implied NACK.

The following is sample code, written in Visual Basic, for the generation of checksums:

```
Public Function ProcessOutputString(outputString As String) As String

Dim i As Integer
Dim CSb1 As Integer
Dim CSb2 As Integer
```

```
Dim CSb3 As Integer
Dim CSb$
Dim X

X = 0
For i = 1 To (Len(outputString))       'Starting with the CMD character
X = X + Asc(Mid(outputString, i, 1))    'adds ascii values together
Next i

CSb1 = 256 - X
CSb2 = 63 And (CSb1)       'Twos Complement
CSb3 = 64 Or (CSb2)        'OR 0x40
CSb$ = Chr(Val("&H" & (Hex(CSb3))))
ProcessOutputString = Chr(2) & outputString & CSb$ & Chr(3)

End Function
```

6.3 COMMAND OVERVIEW

Command Name	<CMD>	<ARG>	RANGE
Program DAC Channel A	10	1-4 ASCII	0-4095
Program DAC Channel B	11	1-4 ASCII	0-4095
Program DAC Channel D	12	1-4 ASCII	0-4095
Program DAC Channel C	13	1-4 ASCII	0-4095
Request DAC A Setpoint	14	None	-
Request DAC B Setpoint	15	None	-
Request DAC D Setpoint	16	None	-
Request DAC C Setpoint	17	None	-
Request Analog Readbacks – J6 Channels 7 – 15	19	None	-
Request Analog Readbacks – J5 Channels 0 – 6	20	None	-
Request HV On Hours Counter	21	None	-
Request Status	22	None	-
Request Software Version	23	None	-
Request Hardware Version	24	None	-
Request Web Server Version	25	None	-
Request Model Number	26	None	-
Reset HV On Hours Counter	30	None	-
Reset Faults	31	None	-
Program Interlock 1	52	1 ASCII	0 or 1
Program Interlock 2	53	1 ASCII	0 or 1

Program Interlock 3	54	1 ASCII	0 or 1
Read Interlock Status	55	None	-
Readback A/D Channel 0 Data	60	None	-
Readback A/D Channel 1 Data	61	None	-
Readback A/D Channel 2 Data	62	None	-
Readback A/D Channel 3 Data	63	None	-
Readback A/D Channel 4 Data	64	None	-
Readback A/D Channel 5 Data	65	None	-
Readback A/D Channel 6 Data	66	None	-
Readback A/D Channel 7 Data	67	None	-
Readback A/D Channel 8 Data	68	None	-
Readback A/D Channel 9 Data	69	None	-
Readback A/D Channel 10 Data	70	None	-
Readback A/D Channel 11 Data	71	None	-
Readback A/D Channel 12 Data	72	None	-
Readback A/D Channel 13 Data	73	None	-
Readback A/D Channel 14 Data	74	None	-
Readback A/D Channel 15 Data	75	None	-
Read Digital Inputs	76	None	-
Program Digital Output Channel 1	84	1 ASCII	0 or 1
Program Digital Output Channel 2	85	1 ASCII	0 or 1
Program Digital Output Channel 3	86	1 ASCII	0 or 1
Program Digital	87	1 ASCII	0 or 1

Output Channel 4			
Program Digital Output Channel 5	88	1 ASCII	0 or 1
Toggle Verbose Mode	92	None	-
Program High Voltage Status	99	1 ASCII	0 or 1

6.5 RESPONSE OVERVIEW

The command responses will follow the same format as outlined above in section 6.1. This list is comprised of Commands with complex responses only. Commands using a simple response will use the <$> character (ASCII 0x24) as a “Success” response or a single character error code. These responses will be eight ASCII characters in length.

Response Name	<CMD>	Response
Request DAC A Setpoint	14	11 ASCII
Request DAC B Setpoint	15	11 ASCII
Request DAC D Setpoint	16	11 ASCII
Request DAC C Setpoint	17	11 ASCII
Request Analog Readbacks – J6	19	24-51 ASCII
Request Analog Readbacks – J5	20	20-41 ASCII
Request Total Hours High Voltage On	21	14 ASCII
Request Status	22	12 ASCII
Request DSP Software Version	23	18 ASCII
Request Hardware Version	24	10 ASCII
Request Web Server Version	25	18 ASCII
Request Model number	26	12 ASCII
Read Interlock Status	55	12 ASCII

Readback A/D Channel 0 Data	60	8-11 ASCII
Readback A/D Channel 1 Data	61	8-11 ASCII
Readback A/D Channel 2 Data	62	8-11 ASCII
Readback A/D Channel 3 Data	63	8-11 ASCII
Readback A/D Channel 4 Data	64	8-11 ASCII
Readback A/D Channel 5 Data	65	8-11 ASCII
Readback A/D Channel 6 Data	66	8-11 ASCII
Readback A/D Channel 7 Data	67	8-11 ASCII
Readback A/D Channel 8 Data	68	8-11 ASCII
Readback A/D Channel 9 Data	69	8-11 ASCII
Readback A/D Channel 10 Data	70	8-11 ASCII
Readback A/D Channel 11 Data	71	8-11 ASCII
Readback A/D Channel 12 Data	72	8-11 ASCII
Readback A/D Channel 13 Data	73	8-11 ASCII
Readback A/D Channel 14 Data	74	8-11 ASCII
Readback A/D Channel 15 Data	75	8-11 ASCII
Read Digital Inputs	76	9 ASCII
Read Digital Output Settings	89	9 ASCII
Program High Voltage Status	99	8 ASCII

6.6 COMMAND STRUCTURE

6.6.1 Program DAC Channel A

Description:
The host requests that the firmware change the setpoint of DAC Channel A.

Direction:
Host to supply

Syntax:
<STX><10><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>10,4095,<CSUM><ETX>

Response:
<STX><10><,><$><,><CSUM><ETX>
<STX><10><,><ARG><,><CSUM><ETX>

where <ARG> = error code

Error Codes TBD, 1=out of range

6.6.2 Program DAC Channel B

Description:
The host requests that the firmware change the setpoint of DAC Channel B.

Direction:
Host to supply

Syntax:
<STX><11><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>11,4095,<CSUM><ETX>

Response:
<STX><11><,><$><,><CSUM><ETX>
<STX><11><,><ARG><,><CSUM><ETX>

where <ARG> = error code

Error Codes TBD, 1=out of range

6.6.3 Program DAC Channel D

Description:
The host requests that the firmware change the setpoint of DAC Channel D.

Direction:
Host to supply

Syntax:
<STX><12><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>12,4095,<CSUM><ETX>

Response:
<STX><12><,><$><,><CSUM><ETX>
<STX><12><,><ARG><,><CSUM><ETX>

where <ARG> = error code

Error Codes TBD, 1 = out of range

6.6.4 Program DAC Channel C

Description:
The host requests that the firmware change the setpoint of DAC Channel C.

Direction:
Host to supply

Syntax:
<STX><13><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>13,4095,<CSUM><ETX>

Response:
<STX><13><,><$><,><CSUM><ETX>
<STX><13><,><ARG><,><CSUM><ETX>

where <ARG> = error code

Error Codes TBD, 1 = out of range

6.6.5 Request DAC A Setpoint

Description:
The host requests that the firmware report the DAC Channel A setpoint.

Direction:
Host to supply

Syntax:
<STX><14><,><CSUM><ETX>

Response:
<STX><14><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>14,4095,<CSUM><ETX>

6.6.6 Request DAC B Setpoint

Description:
The host requests that the firmware report the current DAC Channel B setpoint.

Direction:
Host to supply

Syntax:
<STX><15><,><CSUM><ETX>

Response:
<STX><15><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>15,4095,<CSUM><ETX>

6.6.7 Request DAC D Setpoint

Description:
The host requests that the firmware report the current DAC Channel D setpoint.

Direction:
Host to supply

Syntax:
<STX><16><,><CSUM><ETX>

Response:
<STX><16><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>16,4095,<CSUM><ETX>

6.6.8 Request DAC C Setpoint

Description:
The host requests that the firmware report the current DAC Channel C setpoint.

Direction:
Host to supply

Syntax:
<STX><17><,><CSUM><ETX>

Response:
<STX><17><,><ARG><,><CSUM><ETX>

Where:
<ARG> = 0 - 4095 in ASCII format

Example:
<STX>17,4095,<CSUM><ETX>

6.6.9 Request Analog Readbacks – J6

Description:
The host requests that the firmware transmit the present values of Analog Channels 7 through 15, which are available via connector J6.

Direction:
Host to supply

Syntax:
<STX><19><,><CSUM><ETX>

Example:
<STX>19,<CSUM><ETX>

Response:
<STX><19><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,>
<ARG5><,><ARG6><,><ARG7><,><ARG8><,><ARG9><,><CSUM><ET
X>

Where:
ARGx = 0 - 4095

Example:
<STX>19,4095,4095,4095,4095,4095,4095,4095, 4095, 4095,
<CSUM><ETX>

6.6.10 Request Analog Readbacks – J5

Description:
The host requests that the firmware transmit the present values of Analog Channels 0 through 6, which are available via connector J5.

Direction:
Host to supply

Syntax:
<STX><20><,><CSUM><ETX>

Example:
<STX>20,<CSUM><ETX>

Response:
<STX><20><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,>
<ARG5><,><ARG6><,><ARG7><,><CSUM><ETX>

Where:
ARGx = 0 - 4095

Example:
<STX>20,4095,4095,4095,4095,4095,4095,4095,<CSUM><ETX>

6.6.11 Request Total Hours High Voltage On

Description:
The host requests that the firmware sends the present value of the Total Hours High Voltage On.

Direction:
Host to supply

Syntax:
<STX><21><,><CSUM><ETX>

Example:
<STX>21,<CSUM><ETX>

Response:
<STX><21><,><ARG1>< ARG2>< ARG3><ARG4><ARG5>
<.><ARG6><,><CSUM><ETX>

Where:
<.> = ASCII 0x2E
ARGx = 0 - 9 in ASCII format

Example:
<STX>21,99999.9,<CSUM><ETX>

6.6.12 Request Status

Description:
The host requests that the firmware sends the power supply status.

Direction:
Host to supply

Syntax:
<STX><22><,><CSUM><ETX>

Example:
<STX>22,<CSUM><ETX>

Response:
<STX><22><,><ARG1><,><ARG2><,><ARG3><,><CSUM><ETX>

Where:
<ARG1> 1 = HvOn, 0 = HvOff
<ARG2> 1 = Interlock 1 Open, 0 = Interlock 1 Closed
<ARG3> 1 = Fault Condition, 0 = No Fault

Example:
<STX>22,1,1,<CSUM><ETX>

NOTE: This response will also be sent in an unsolicited manner when a change of state is detected on the HvOn and Interlock 1 bits.

6.6.13 Request DSP Software Part Number/Version

Description:
The host requests that the firmware sends the DSP firmware version.

Direction:
Host to supply

Syntax:
<STX><23><,><CSUM><ETX>

Example:
<STX>23,<CSUM><STX>

Response:
<STX><23><,>< ARG><,><CSUM><ETX>

Where:
<ARG> consists of eleven ASCII characters representing the current firmware part number/version. The format is SWM9999-999

Example:
<STX>23,SWM9999-999,<CSUM><ETX>

6.6.14 Request Hardware Version

Description:
The host requests that the firmware sends the hardware version.

Direction:
Host to supply

Syntax:
<STX><24><,><CSUM><ETX>

Example:
<STX>24,<CSUM><ETX>

Response:
<STX><24><,>< ARG><,><CSUM><ETX>

Where:
<ARG> consists of 3 ASCII characters representing the hardware version. The format is ANN, where A is an alpha character and N is a numeric character

Example:
<STX>24,A01,<CSUM><ETX>

6.6.15 Request Webserver Software Part Number/Version

Description:
The host requests that the firmware sends the Web Server firmware part number/version.

Direction:
Host to supply

Syntax:
<STX><25><,><CSUM><ETX>

Example:
<STX>25,<CSUM><ETX>

Response:
<STX><25><,><ARG><,><CSUM><ETX>

Where:
<ARG> consists of eleven ASCII characters representing the current firmware part number/version. The format is SWM9999-999

Example:
<STX>25,SWM9999-999,<CSUM><ETX>

6.6.16 Request Model Number

Description:
The host requests that the firmware sends the unit model number

Direction:
Host to supply

Syntax:
<STX><26><,><CSUM><ETX>

Example:
<STX>26,<CSUM><ETX>

Response:
<STX><26><,><ARG><,><CSUM><ETX>

Where:
<ARG> consists of five ASCII characters representing the model number.
The format is XNNNN, where N is a numeric character.

Example:
<STX>25,X9999,<CSUM><ETX>

6.6.17 Reset Run Hours

Description:
The host requests that the firmware resets the run hour counter.

Direction:
Host to supply

Syntax:
<STX><30><,><CSUM><ETX>

Example:
<STX>30,<CSUM><ETX>

Response:
<STX><30><,><$><,><CSUM><ETX>

6.6.18 Reset Faults

Description:
The host requests that the firmware resets all Fault messages and indicators.

Direction:
Host to supply

Syntax:
<STX><31><,><CSUM><ETX>

Example:
<STX>31,<CSUM><ETX>

Response:
<STX><31><,><$><,><CSUM><ETX>

6.6.19 Program Interlock State

Description:
The host requests that the firmware Program the state of a specific Interlock Channel.

Direction:
Host to supply

Syntax:
<STX><CH><,><ARG><,><CSUM><ETX>
Where CH is the command for a specific Interlock Channel, and ARG is a 1 or a 0 to set or clear the interlock.

Channel Number	Command
Interlock 1:	52
Interlock 2:	53
Interlock 3:	54

Response:
<STX><CH><,><$><,><CSUM><ETX>

Example:
<STX>52,1,<CSUM><ETX>
Where 1 signifies that interlock channel 1 has been energized.

6.6.20 Read Interlock Status

Description:
The host requests that the firmware read the status of all interlock channels.

Direction:
Host to supply

Syntax:
<STX><55><,><CSUM><ETX>

Response:
<STX><55><,><ARG1><,><ARG2><,><ARG3><,><CSUM><ETX>
Where ARG1 through ARG3 are Interlocks 1 through 3. A 1 indicates that the Interlock is energized

Example:
<STX>55,<CSUM><ETX>

6.6.21 Readback A/D Channel Data

Description:
The host requests that the firmware report data from a specific Analog Channel.

Direction:
Host to supply

Syntax:
<STX><CH><,><CSUM><ETX>
Where CH is the command representing a specific A/D Channel:

Channel Number	Command	Channel Number	Command
Channel 0:	60	Channel 8:	68
Channel 1:	61	Channel 9:	69
Channel 2:	62	Channel 10:	70
Channel 3:	63	Channel 11:	71
Channel 4:	64	Channel 12:	72
Channel 5:	65	Channel 13:	73
Channel 6:	66	Channel 14:	74
Channel 7:	67	Channel 15:	75

Response:
<STX><CH><,><ARG><,><CSUM><ETX>

Where:
<ARG>=0-4095 in ASCII format representing unscaled A/D Channel data.

Example:
<STX>68,4095,<CSUM><ETX>

Note:
Channel 0 is the Ambient Temperature Monitor and Channel 1 is the S.I.C. Board Power Supply Monitor.

6.6.22 Read Digital Input Status

Description:
The host requests that the firmware report the current status of the digital inputs.

Direction:
Host to supply

Syntax:
<STX><76><,><CSUM><ETX>

Response:
<STX><76><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,><ARG5>
<,><ARG6><,><ARG7><,><ARG8><,><CSUM><ETX>

Where:
<ARGn> = ASCII Characters 1 or 0 (0x31 or 0x30) representing Digital Input Channel data. Digital Input Channel 1 is represented by ARG1

Example:
<STX>76,1,1,1,1,1,1,1,1,<CSUM><ETX>
Where all input channels are detecting binary 1s.

6.6.23 Program a Digital Output Channel

Description:
The host requests that the firmware SET or CLEAR a Digital Output Channel.

Direction:
Host to supply

Syntax:
<STX><CH><,><ARG><,><CSUM><ETX>
Where CH is the command for a specific Digital Output Channel, and ARG is a 1 or a 0 to set or clear the output.

Channel Number	Command
Channel 1:	84
Channel 2:	85
Channel 3:	86
Channel 4:	87
Channel 5:	88

Response:
<STX><CH><,><$><,><CSUM><ETX>

Example:
<STX>86,1,<CSUM><ETX>
The above command sets output channel 3 to 1.

6.6.24 Read Digital Output Settings

Description:
The host requests that the firmware report the current status of the digital outputs.

Direction:
Host to supply

Syntax:
<STX><89><,><CSUM><ETX>

Response:
<STX><89><,><ARG1><,><ARG2><,><ARG3><,><ARG4><,><ARG5>
<,><CSUM><ETX>

Where:
<ARGn> = ASCII Characters 1 or 0 (0x31 or 0x30) representing Digital Input Channel data. Digital Output Channel 1 is represented by ARG1.

Example:
<STX>89,1,1,1,1,1,<CSUM><ETX>
Where all output channels are set to 1s.

6.6.25 Toggle Verbose Mode

Description:
The host requests that the firmware provide or cease continuous data updates via the current communications channel. Transmit this command to begin the transmission of data, transmit again to stop transmission.

Direction:
Host to supply

Syntax:
<STX><92><,><CSUM><ETX>

Response:
<STX><92><,><$><,><CSUM><ETX>

Example:
<STX>92,<CSUM><ETX>

6.6.26 Program High Voltage On/Off

Description:
The host requests that the firmware to turn on or off High Voltage.

Direction:
Host to supply

Syntax:
<STX><99><,><ARG><,><CSUM><ETX>

Where:
<ARG> 1 = On, 0 = Off in ASCII format

Example:
<STX>99,1,<CSUM><ETX>

Response:
<STX><99><,><$><,><CSUM><ETX>
<STX><99><,><ARG><,><CSUM><ETX>

where <ARG> = error code

Error Codes TBD,
1 = out of range
2 = Interlock 1 open, High Voltage Disabled

6.7 SPELLMAN TEST COMMANDS

- Program Hardware Version (Hardware setup)
- Set USB Mode (Program USB)
- Set USB Page Address (Program USB)
- Send USB Page Data (Program USB)
- Toggle Passthrough Mode (Diagnostics)
- Store A/D Calibration Value (Hardware setup)

Contact Spellman High Voltage for details and the syntax of these commands.

6.8 SERIAL COMMAND HANDLING

6.8.1 Command Time Out

The host computer should set a serial time out at approximately 100mS. This allows the DSP to process the incoming message, and transmit a response. The DSP will initiate a reply to incoming messages in approximately 1-2mS, with a worst case of 5mS.

6.8.2 Buffer Flushing

The DSP will flush the incoming serial data buffer every time an STX is received. This provides a mechanism to clear the receive buffer of partial or corrupt messages.

6.8.3 Handshaking

The only handshaking implemented on the host interface, is built in to the implementation of this protocol. That is, the host must initiate all communications. If the supply receives a program command, an acknowledge message is sent back to the host via the “$” message. If the host does not receive an acknowledge within the time out window, the host should consider the message lost or the device off-line.

Similarly, if the supply receives a request command, the requested data is sent back to the host. If the host does not receive the requested data within the time out window, the host should consider the message lost or the device off-line.

This essentially uses the full-duplex channel in a half-duplex communication mode.

7.0 S.I.C. Board Resource Utilization Table

Analog to Digital Converter Resources

A/DC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
ANA 0	N/A	0.0732 V/°C	N/A: SIC Temperature Sensor
ANA 1	N/A	10.5 mV	N/A: SIC + 24 volt Monitor
ANA 2	JB5-6		
ANA 3	JB5-8		
ANA 4	JB5-12		
ANA 5	JB5-10		
ANA 6	JB5-2		
ANA 7	JB6-1		
ANA 8	JB6-3		
ANA 9	JB6-2		
ANA 10	JB6-5		
ANA 11	JB6-4		
ANA 12	JB6-7		
ANA 13	JB6-6		
ANA 14	JB6-9		
ANA 15	JB6-8		

Digital to Analog Converter Resources

DAC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
Channel A	JB5-1		
Channel B	JB5-5		
Channel C	JB5-3		
Channel D	JB5-4		

Digital Input Resources

Input Channel	Header Location	Project Signal Name
IN 1 – LS, IRQ	JB7-1	
IN 2 – LS, IRQ	JB7-2	
IN 3 – LS, IRQ	JB7-3	
IN 4 – LS, IRQ	JB7-4	
IN 5 – LS, IRQ	JB7-5	
IN 6 – LS, IRQ	JB7-6	
IN 7 – LS, IRQ	JB7-7	
IN 8 - IRQ	JB7-8	

LS = Level Sensitive. IRQ = Able to send an Interrupt to the DSP

Digital Output Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
OUT 1	JB8-1		
OUT 2	JB7-12		
OUT 3	JB7-11		
OUT 4	JB7-10		
OUT 5	JB7-9		

Interlock Resources

Interlock Block	Header Location	Initial Power Up State	Project Signal Name
Interlock 1 - NC	JB5-11		
Interlock 1 - NO	JB5-13		
Interlock 1 – COM	JB5-14		
Interlock 1 - AUX	JB8-10		
Interlock 2 - NC	JB8-8		
Interlock 2 – NO	JB8-6		
Interlock 2 – COM	JB8-7		
Interlock 2 - AUX	JB8-9		
Interlock 3 - NC	JB8-4		
Interlock 3 - NO	JB8-2		
Interlock 3 - COM	JB8-3		
Interlock 3 - AUX	JB8-5		

8.0 Product Specific Usage

Note: Commands that differ in nomenclature from the specification will be cross-referenced to their equivalent. Commands that do not differ will simply be listed.

8.1 MNX50P50

Note: Tabular information applies to the Standard Model with the 10 Volt DAC Reference and the X3366 Model with the 5 Volt Reference except as noted.

Analog to Digital Converter Resources

A/DC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
ANA 0	N/A	0.0732 V/°C	SIC Temperature Sensor: 10 mV/ Degree C
ANA 1	N/A	10.5 mV	SIC + 24 volt Monitor
ANA 2	JB5-6	12.21 V	HV_MONITOR: 0-5V = 0-50kV
ANA 3	JB5-8	586.1 uA	BEAM_CURRENT: 0-5V = 0-2mA
ANA 4	JB5-12	879 uA	FIL_CURRENT: 1V = 1A
ANA 5	JB5-10	1.343 mV	FIL_VOLTAGE: 5V / 3A
ANA 6	JB5-2	0.0732 V/°C	MULT_TEMP: 10 mV/ Degree C

Digital to Analog Converter Resources: 10 Volt Reference (5 Volt Reference)

DAC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
Channel A	JB5-1	12.21 V	KV_PROG
Channel B	JB5-5	488 uA	MA_PROG
Channel C	JB5-3	2.442 mA (1.221 mA)	FIL_LIMIT_PROG
Channel D	JB5-4	2.442 mA (1.221 mA)	FIL_PREHEAT

Interlock Resources

Interlock Block	Header Location	Initial Power Up State	Project Signal Name
Interlock 1 - NC	JB5-11		
Interlock 1 - NO	JB5-13		INTRLK_RTN
Interlock 1 – COM	JB5-14	Off	INTRLK
Interlock 1 - AUX	JB8-10		

Product Specific Command	Generic Command	Comments
Program Voltage Setpoint	Program DAC Channel A	LSB = 12.21v
Program Emission Current Setpoint	Program DAC Channel B	LSB = 488 uA
Program Filament Pre-heat Setpoint	Program DAC Channel D	LSB = 1.221 mA (Vref= 5 v) LSB = 2.442 mA (Vref=10v)
Program Filament Current Limit	Program DAC Channel C	LSB = 1.221 mA (Vref= 5 v) LSB = 2.442 mA (Vref=10v)
Request Voltage Setpoint	Request DAC A Setpoint	LSB = 12.21 v
Request Emission Current Setpoint	Request DAC B Setpoint	LSB = 488 uA
Request Filament Pre-heat Setpoint	Request DAC D Setpoint	LSB = 1.221 mA (Vref= 5 v) LSB = 2.442 mA (Vref=10v)
Request Filament Current Limit	Request DAC C Setpoint	LSB = 1.221 mA (Vref= 5 v) LSB = 2.442 mA (Vref=10v)
Request Analog Readbacks	Request Analog Channels – J5	Ch 2: LSB = 12.21 V Ch 3: LSB = 586.1 uA Ch 4: LSB = 879 uA Ch 5: LSB = 1.343mV Ch 6: LSB = 0.0732°
	Request “HV On” Hours Counter	
	Request Status	
	Request Software Version	
	Request Hardware Version	
	Request Web Server Version	
	Request Model Number	
	Reset “HV On” Hours Counter	
	Request Network Settings	
	Program Network Settings	
	Program High Voltage Status	

8.2 SL80PN1200 (X3442)

Analog to Digital Converter Resources

A/DC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
ANA 0	N/A	0.0732 V/°C	N/A: SIC Temperature Sensor
ANA 1	N/A	10.5 mV	N/A: SIC + 24 volt Monitor
ANA 2	JB5-6	39.072 V	HV Monitor
ANA 3	JB5-8	2.44 mA	Emission Current Monitor
ANA 6	JB5-2	0.0732 V/°C	Multiplier Temperature

Digital to Analog Converter Resources: 10 Volt Reference

DAC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
Channel A	JB5-1	39.072 V	KV_PROG
Channel B	JB5-5	2.44 mA	MA_PROG
Channel C	JB5-3	.0016 mA	FIL_PRE-HEAT_PROG

Digital Output Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
OUT 1	JB8-1	CLEAR	FAULT_RESET

Product Specific Command	Generic Command	Comments
Program Voltage Setpoint	Program DAC Channel A	LSB = 39.072 V
Program Emission Current Setpoint	Program DAC Channel B	LSB = 2.44 mA
Program Filament Pre-heat Setpoint	Program DAC Channel C	LSB = .0016 mA (Vref=10v)
Request Voltage Setpoint	Request DAC A Setpoint	LSB = 39.072 V
Request Emission Current Setpoint	Request DAC B Setpoint	LSB = 2.44 mA
Request Analog Readbacks	Request Analog Channels – J5	Ch 2: LSB = 39.072 V Ch 3: LSB = 2.44 mA Ch 6: LSB = 0.0732°
	Request Status	
	Request Software Version	
	Request Hardware Version	
	Request Web Server Version	
	Request Model Number	
Product Specific Command	Generic Command	Comments
Reset Faults	Program Digital Output Channel 1	SET = Reset CLEAR = Normal
	Request Network Settings	
	Program Network Settings	
	Program High Voltage Status	

8.3 XLG130P1200 X3459

Analog to Digital Converter Resources

A/DC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
ANA 0	N/A	0.0732 V/°C	N/A: SIC Temperature Sensor
ANA 1	N/A	10.5 mV	N/A: SIC + 24 volt Monitor
ANA 2	JB5-6	31.746 V	HV Monitor
ANA 3	JB5-8	2.44 mA	Emission Current Monitor

Digital to Analog Converter Resources: 10 Volt Reference

DAC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
Channel A	JB5-1	31.746 V	KV_PROG
Channel B	JB5-5	2.44 mA	MA_PROG

Digital Input Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
IN 1	JB7-1	NA	FAULT_N (0 = fault)
IN 2	JB7-2	NA	INTERLOCK (0 = closed)
IN 3	JB7-3	NA	FILAMENT (15V = small, 0 = large)

Digital Ouput Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
OUT 1	JB8-1	CLEAR	FILAMENT_SELECT

Interlock Resources

Interlock Block	Header Location	Initial Power Up State	Project Signal Name
Interlock 1 - NC	JB5-11		
Interlock 1 - NO	JB5-13		Ground
Interlock 1 – COM	JB5-14	OPEN (not energized)	RESET FAULT
Interlock 1 - AUX	JB8-10		
Interlock 2 – NC	JB8-8		
Interlock 2 – NO	JB8-6		HV ON 2
Interlock 2 – COM	JB8-7	OPEN (not energized)	HV ON 1
Interlock 2 - AUX	JB8-9		
Interlock 3 – NC	JB8-4		HV OFF 2
Interlock 3 – NO	JB8-2		
Interlock 3 – COM	JB8-3	OPEN (not energized)	HV OFF 1
Interlock 3 - AUX	JB8-5		

Product Specific Command	Generic Command	Comments
Program Voltage Setpoint	Program DAC Channel A	LSB = 31.746 V
Program Emission Current Setpoint	Program DAC Channel B	LSB = 2.44 mA
Request Voltage Setpoint	Request DAC A Setpoint	LSB = 31.746 V
Request Emission Current Setpoint	Request DAC B Setpoint	LSB = 2.44 mA
Request Analog Readbacks	Request Analog Channels – J5	Ch 2: LSB = 31.746 V Ch 3: LSB = 2.44 mA Ch 6: LSB = 0.0732°
	Request Status	
	Request Software Version	
	Request Hardware Version	
	Request Web Server Version	
	Request Model Number	
Filament Select	Program Digital Output Channel 1	SET = Small CLEAR = Large
	Request Network Settings	
	Program Network Settings	
	Program High Voltage Status	
Reset Fault	Program Interlock State Interlock 1	Pulse Activated (> 1 ms) SET = reset CLEAR = not reset
HV ON	Program Interlock State Interlock 2	Pulse Activated (> 1 ms) SET then CLEAR
HV OFF	Program Interlock State Interlock 3	Pulse Activated (> 1 ms) SET then CLEAR Note: HV ON must be in CLEAR state
Request Filament Status	Read Digital Input Status	<ARG1> = FAULT <ARG2> = INTERLOCK_N <ARG3> = FILAMENT_N all other arguments are NA

8.4 XLG80P800 X3461

Analog to Digital Converter Resources

A/DC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
ANA 0	N/A	0.0732 V/°C	N/A: SIC Temperature Sensor
ANA 1	N/A	10.5 mV	N/A: SIC + 24 volt Monitor
ANA 2	JB5-6	19.536 V	HV Monitor
ANA 3	JB5-8	2.44 mA	Emission Current Monitor

Digital to Analog Converter Resources: 10 Volt Reference

DAC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
Channel A	JB5-1	19.536 V	KV_PROG
Channel B	JB5-5	2.44 mA	MA_PROG

Digital Input Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
IN 1	JB7-1	NA	FAULT_N (0 = fault)
IN 2	JB7-2	NA	INTERLOCK (0 = closed)
IN 3	JB7-3	NA	FILAMENT (15V = small, 0 = large)

Digital Ouput Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
OUT 1	JB8-1	CLEAR	FILAMENT_SELECT

Interlock Resources

Interlock Block	Header Location	Initial Power Up State	Project Signal Name
Interlock 1 - NC	JB5-11		
Interlock 1 - NO	JB5-13		Ground
Interlock 1 – COM	JB5-14	OPEN (not energized)	RESET FAULT
Interlock 1 - AUX	JB8-10		
Interlock 2 – NC	JB8-8		
Interlock 2 – NO	JB8-6		HV ON 2
Interlock 2 – COM	JB8-7	OPEN (not energized)	HV ON 1
Interlock 2 - AUX	JB8-9		
Interlock 3 – NC	JB8-4		HV OFF 2
Interlock 3 – NO	JB8-2		
Interlock 3 – COM	JB8-3	OPEN (not energized)	HV OFF 1
Interlock 3 - AUX	JB8-5		

Product Specific Command	Generic Command	Comments
Program Voltage Setpoint	Program DAC Channel A	LSB = 19.536 V
Program Emission Current Setpoint	Program DAC Channel B	LSB = 2.44 mA
Request Voltage Setpoint	Request DAC A Setpoint	LSB = 19.536 V
Request Emission Current Setpoint	Request DAC B Setpoint	LSB = 2.44 mA
Request Analog Readbacks	Request Analog Channels – J5	Ch 2: LSB = 19.536 V Ch 3: LSB = 2.44 mA Ch 6: LSB = 0.0732°
	Request Status	
	Request Software Version	
	Request Hardware Version	
	Request Web Server Version	
	Request Model Number	
Filament Select	Program Digital Output Channel 1	SET = Small CLEAR = Large
	Request Network Settings	
	Program Network Settings	
	Program High Voltage Status	
Reset Fault	Program Interlock State Interlock 1	Pulse Activated (> 1 ms) SET = reset CLEAR = not reset
HV ON	Program Interlock State Interlock 2	Pulse Activated (> 1 ms) SET then CLEAR
HV OFF	Program Interlock State Interlock 3	Pulse Activated (> 1 ms) SET then CLEAR Note: HV ON must be in CLEAR state
Request Filament Status	Read Digital Input Status	<ARG1> = FAULT <ARG2> = INTERLOCK_N <ARG3> = FILAMENT_N all other arguments are NA

8.5 SL6PN1200 X3496

A/DC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
ANA 0	N/A	0.0732 V/°C	N/A: SIC Temperature Sensor
ANA 1	N/A	10.5 mV	N/A: SIC + 24 volt Monitor
ANA 2	JB5-6	1.4625 V	HV Monitor
ANA 3	JB5-8	0.0488 mA	Emission Current Monitor
ANA 8	JP5-1,JP5-2	0.806 mV	Constant 1.981 Volts
ANA 9	JP4-1,JP4-2	0.806 mV	Constant 0.824 Volts

Analog to Digital Converter Resources

Digital to Analog Converter Resources: 10 Volt Reference

DAC Channel	Header Location	Scaling (LSB Multiplier)	Project Signal Name
Channel A	JB5-1	1.4625 V	KV_PROG
Channel B	JB5-5	0.04884 mA	MA_PROG

Digital Input Resources

Input Channel	Header Location	Initial Power Up State	Project Signal Name
IN 1	JB7-1	NA	FAULT_N (0 = fault @ input to U11)
IN 2	JB7-2	NA	INTERLOCK (0 = closed @ input to U11)
IN 3	JB7-3	NA	
IN 4	JB7-4	NA	OVER VOLTAGE (1 = Voltage Mode @ input to U11)
IN 5	JB7-5	NA	OVER CURRENT (0 = HV is On @ input to U11)
IN 6	JB7-6	NA	REG ERROR (1 = PS is On @ input to U11)
IN 7	JB7-7	NA	ARC
IN 8	JB7-8	NA	TEMP

Digital Ouput Resources

Output Channel	Header Location	Initial Power Up State	Project Signal Name
OUT 1	JB8-1	CLEAR	NA
OUT 2	JB7-12	CLEAR	LOCAL/REMOTE

Interlock Resources

Interlock Block	Header Location	Initial Power Up State	Project Signal Name
Interlock 1 - NC	JB5-11		
Interlock 1 - NO	JB5-13		GROUND
Interlock 1 – COM	JB5-14	OPEN (not energized & connected to JB5-11 thru relay contact)	RESET_N
Interlock 1 - AUX	JB8-10		
Interlock 2 – NC	JB8-8		
Interlock 2 – NO	JB8-6		HV ON 2
Interlock 2 – COM	JB8-7	OPEN (not energized)	HV ON 1
Interlock 2 - AUX	JB8-9		
Interlock 3 – NC	JB8-4		HV OFF 2
Interlock 3 – NO	JB8-2		
Interlock 3 – COM	JB8-3	OPEN (not energized)	HV OFF 1
Interlock 3 - AUX	JB8-5		

Product Specific Command	Generic Command	Comments
Program Voltage Setpoint	Program DAC Channel A	LSB = 1.4652 V
Program Emission Current Setpoint	Program DAC Channel B	LSB = 0.04884 mA
Request Voltage Setpoint	Request DAC A Setpoint	LSB = 1.4652 V
Request Emission Current Setpoint	Request DAC B Setpoint	LSB = 0.04884 mA
Request Analog Readbacks	Request Analog Channels – J5	Ch 2: LSB = 1.4652 V Ch 3: LSB = 0.04884 mA Ch 6: LSB = 0.0732°
	Request Status	
	Request Software Version	
	Request Hardware Version	
	Request Web Server Version	
	Request Model Number	
	Program Digital Output Channel 1	
Local/Remote Select	Program Digital Output Channel 2	SET = Remote CLEAR = Local
	Request Network Settings	
	Program Network Settings	
	Program High Voltage Status	
Reset Fault	Program Interlock State Interlock 1	Pulse Activated (> 1 ms) SET = reset CLEAR = not reset
HV ON	Program Interlock State Interlock 2	Pulse Activated (> 1 ms) SET then CLEAR

HV OFF	Program Interlock State Interlock 3	Pulse Activated (> 1 ms) SET then CLEAR Note: HV ON must be in CLEAR state
Request Digital Status	Read Digital Input Status	<ARG1> = FAULT (0 = Fault) <ARG2> = INTERLOCK Open (1 = Interlock Open) <ARG3> = Local/Remote (1 = Remote, 0 = Local) <ARG4> = OVERVOLTAGE (0 = Over Voltage Fault) <ARG5> = OVERCURRENT (0 = Over Current Fault) <ARG6> = REG ERROR (0 = Reg Error) <ARG7> = ARC (0 = Arc Error) <ARG8> = TEMP (0 = Temperature Error)